

SPACE BYTETM 8085 CPU

PRODUCT SUPPORT DOCUMENTATION

SPACE BYTE COMPUTER CORP.
6464 Sunset Blvd. Suite 530
Los Angeles, CA 90028
(213) 468-8080

COPYRIGHT DECLARATION

Any information hereunder contained, in the form of text, source listings, photographs, schematic or block diagrams, is the sole property of the SPACE BYTE CORP. and may not be reproduced or duplicated in any manner, or on any medium now known or yet to be devised without the written consent of the SPACE BYTE CORP.

© SPACE BYTE CORPORATION, 1977

ENGINEERING ERRATA

Caution: Always open iCOM drive doors prior to depressing reset or disk crash will result.

* * *

NOTE:

NEVER ATTEMPT TO INSERT OR REMOVE A CARD
FROM THE MOTHERBOARD WITH THE POWER TURNED ON.

* * *

SPACE BYTE 8085 CPU

Introduction and Design Philosophy

The design and marketing philosophy behind the development of the SPACE BYTE 8085 CPU was to create a single board computer containing all of the necessary I/O to support a disk based small business computer system. Additionally, the CPU card, with the optional SPACE BYTE 2708/2716 EPROM Programmer attached will function as the heart of a software development system- utilizing the CPU card as a self contained dedicated controller.

The ICOM FD3700, or frugal floppy with the FC-360 controller was chosen to interface directly with the SPACE BYTE 8085 CPU for the following reasons:

- 1) Simplicity of interface- connection directly to the CPU card, eliminating the need for the S-100 interface card (all software for disk I/O handling contained in the on board 3K system monitor.
- 2) Full size, 8" IBM compatible 256K diskette.
- 3) Full software support- both FDOS III relocatable assembler and operating system and DEBBI (Disk Extended Basic) are available from ICOM assembled for the SPACE BYTE 8085 CPU hardware configuration. Absolutely no patching is necessary.
- 4) ICOM products are well supported at the retail and OEM levels.

With this philosophy in mind, the SPACE BYTE 8085 CPU was further developed to allow more versatility. This ofcourse being the inclusion of the necessary routines to support the Tarbell cassette medium and Polymorphic Video Interface, enabling the use of the Electric Pencil, by Michael Shroyer for complete word processor capability. A variety of jumper schemes allow such options as relocatable system PROM address (0000, C000, or E000), use of the INTEL 2716 EPROM, for up to 6K of onboard application firmware and pin 4 & 5 clear to send function for serial ports and M-WRITE generation, through the bus.

Careful review of the following information will assure complete utilization of the SPACE BYTE 8085 CPU.

SPACE BYTE 8085 JUMPER OPTIONS

The SPACE BYTE 8085 CPU card is designed to be useful in many applications. 18 option jumper positions have been included to make the job of configuring the card for your application easier. The card is shipped with 8 of these jumpers etched in place, representing the most common S-100 bus application. The card should work in your system with little or no changes necessary.

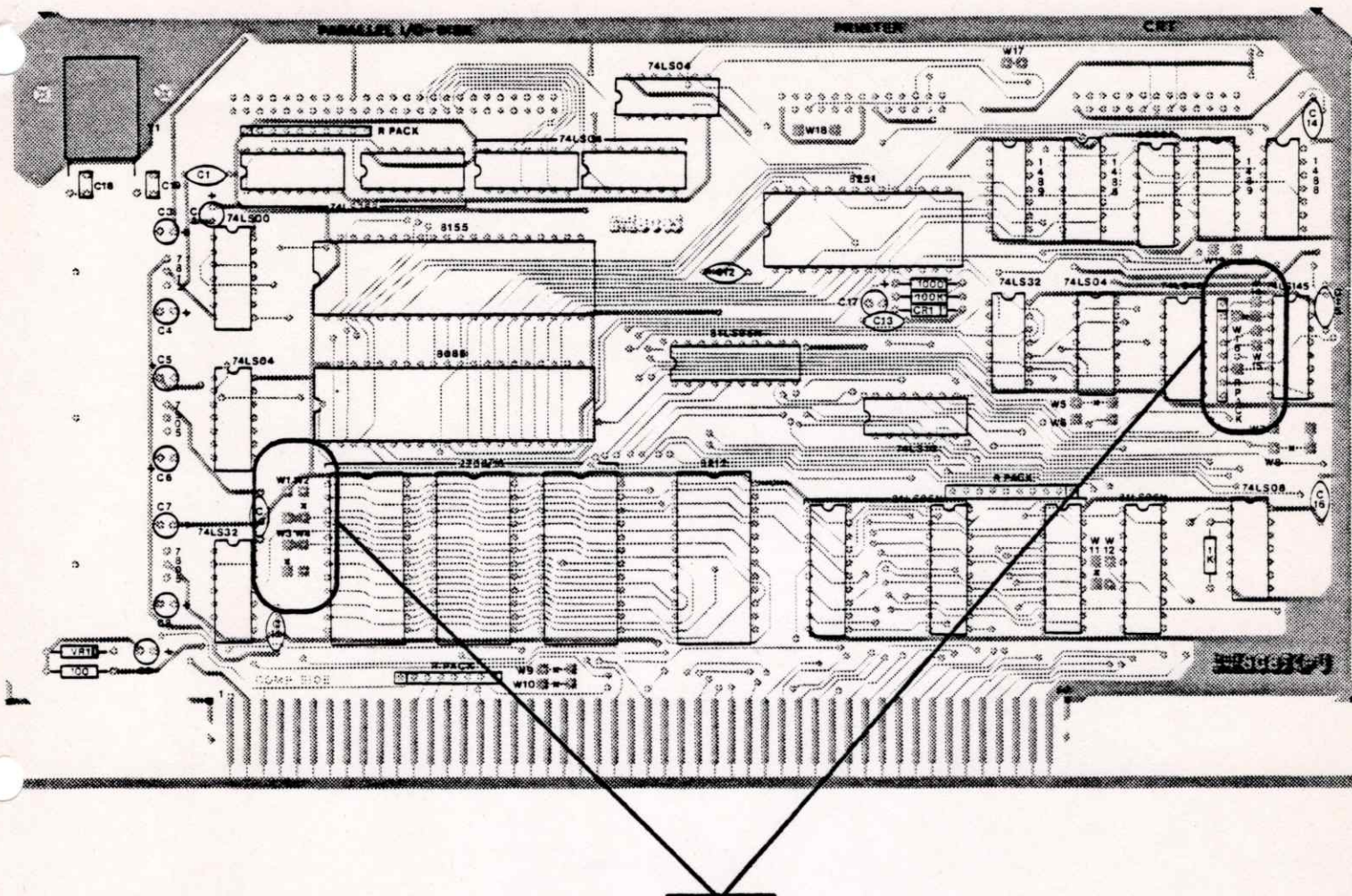
The diagrams in the following pages explain the functions of each jumper. Keep in mind that changing jumpers may require a firmware change. For example, if a PROM starting address of E000 is desired, the 3K SPACE BYTE PROM Monitor will not function because it has been assembled for operation at a starting address of C000. All links to the SPACE BYTE version of ICOM Software requires that the PROM monitor start at address C000.

The card can be strapped for starting addresses of E000 and 0000, but the user is then responsible for providing the necessary software for the applications. The jumpers that can be added or deleted without requiring a software change are:

W-10
W-9
W-11
W-12
W-18
W-19

As shipped from SPACE BYTE CORPORATION, the 8085 CPU card has a starting address of C000. When the hardware system reset switch is depressed, the CPU card jumps automatically to this address. The port addressing and memory map for all three configurations possible are shown below.

0000 START	C000 START	E000 START	FUNCTION
0000-0BFF	C000-CBFF	E000-EBFF	3K SPACE BYTE PROM MONITOR
0000-17FF	C000-D7FF	E000-F7FF	PROM MEMORY AREA USING 2K 2716 DEVICES
1800-18FF	D800-D8FF	F800-F8FF	256 BYTE SCRATCH RAM (in the 8155)
0	C0	E0	USART DATA PORT I/O
1	C1	E1	USART STATUS CONTROL PORT I/O
18	D8	F8	8155 CONTROL STATUS PORT
19	D9	F9	8155 PORT A (8-BITS)
1A	DA	FA	8155 PORT B (8-BITS)
1B	DB	FB	8155 PORT C (6-BITS)
1C	DC	FC	8155 TIMER PORT (LO-ORDER)
1D	DD	FD	8155 TIMER PORT (HI-ORDER)
C	CC	EC	8-BIT UNLATCHED INPUT PORT
NA	6E	NA	MAG TAPE CONTROL PORT
NA	6F	NA	MAG TAPE DATA PORT
NA	F8	NA	VIDEO CARD KEYBOARD PORT
NA	F800-FFF	NA	VIDEO CARD RAM MEMORY (DISPLAY MEMORY)



PROM STRAPPING OPTIONS

W1

This jumper is not installed. W1 conflicts with W2 and only one can be installed. W1 connects +5VDC to pin 21 of all 3 PROM sockets. This is needed when using the INTEL 2716 PROM device.

W2

This jumper is installed. W2 conflicts with W1 and only one can be installed. W2 connects -5VDC to pin 21 of all 3 PROM sockets. This is needed when using 2708 PROM devices in the card.

W3

This jumper is installed. W3 conflicts with W4 and only one can be installed. W3 connects +12VDC to pin 19 of all 3 PROM sockets. This is needed when using 2708 PROM devices in the card.

W4

This jumper is not installed. W4 conflicts with W3 and only one can be installed. W4 connects A10 to pin 19 of all 3 PROM sockets. This is needed when using INTEL 2716 PROM devices on the card.

W14, W15, W16

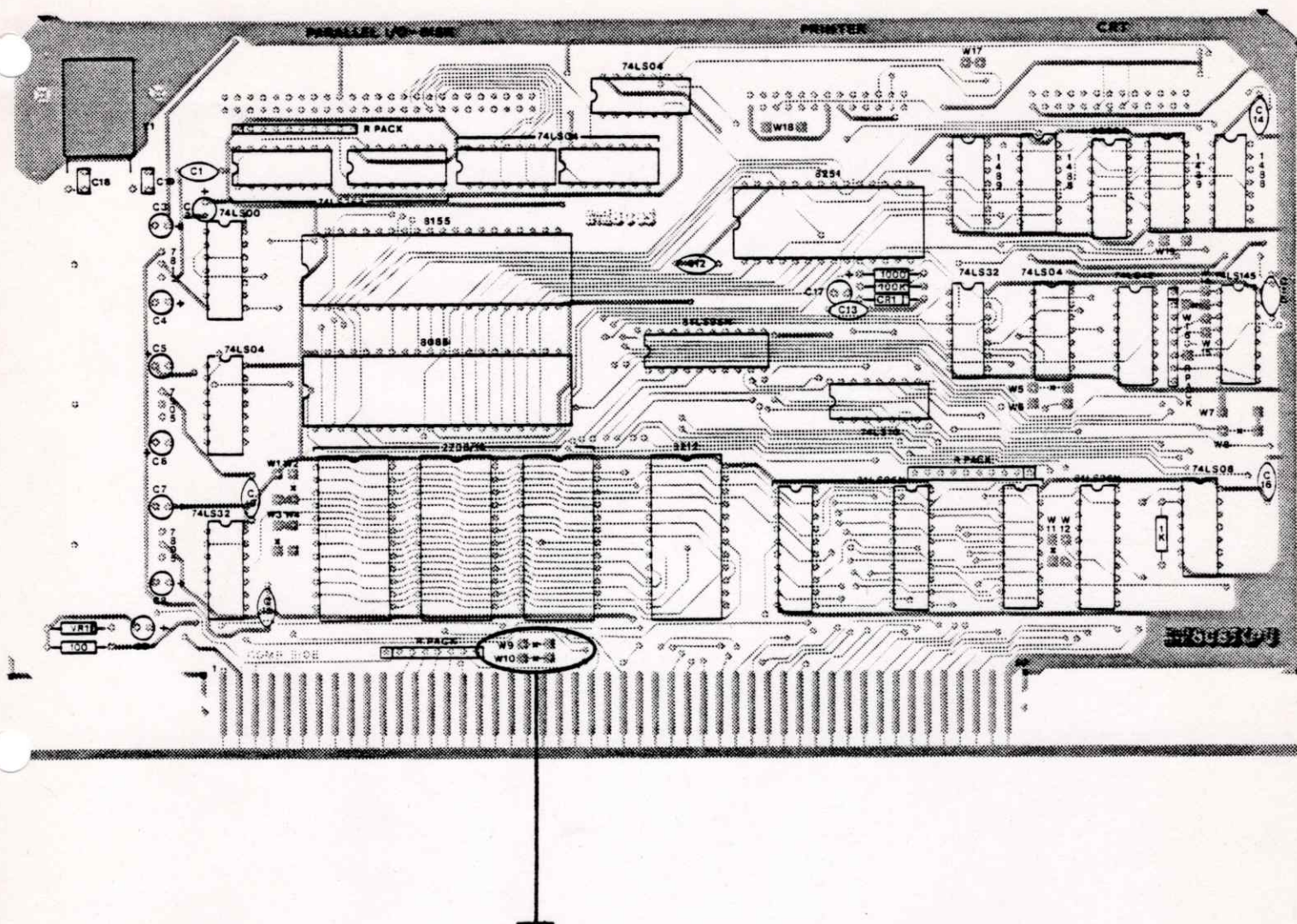
These three jumpers are not installed. W14 has a horizontal jumper etched, connecting to the top of W16. This connection is required when using 2708 PROM devices. If 2716 PROM devices are to be used, cut this etch, and install three vertically positioned jumpers in W-14, W-15, and W-16. When these jumpers are installed the PROM address space is 6K. Refer to the memory map for absolute address ranges.

W7

This jumper is not installed. W7 conflicts with W8 and only one can be installed. W7 sets the starting address of the card to 0000, and limits the addressing range to 8K total. If 6K bytes of PROM is installed, only the scratch RAM and 1K of additional off-board RAM can be controlled. This RAM must be addressed at 1C00-1FFF. This is the typical case for single card dedicated controller applications.

W8

This jumper is installed. W8 conflicts with W7 and only one can be installed. W8 places board starting address under the control of W5 and W6. This jumper should only be removed if the 8085 CPU is to be used in single board controller applications.



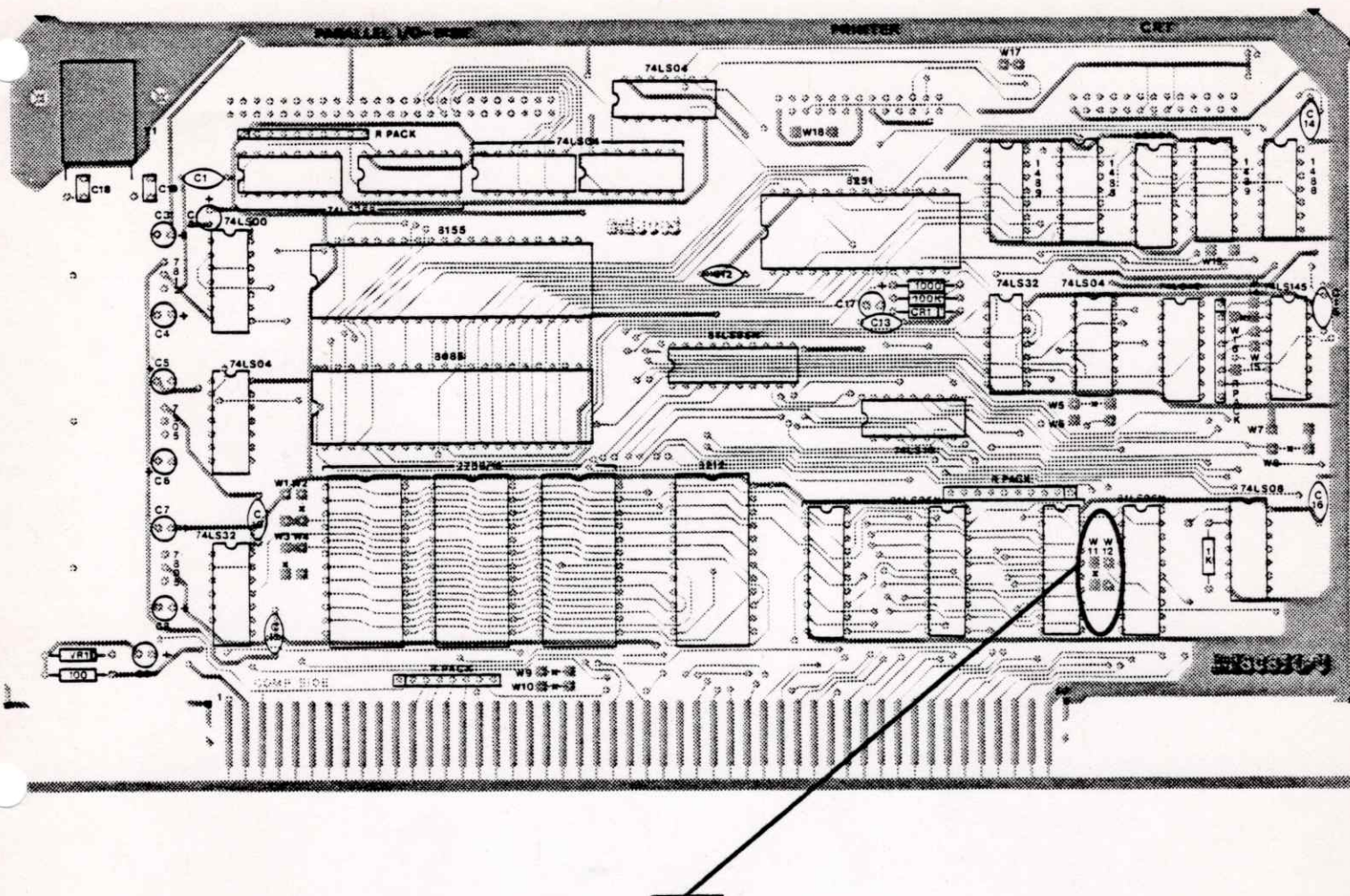
S-100 BUS OPTIONS

W9

This jumper is installed. This jumper connects the jump-start bistable output to pin 67 of the bus. This pin is low (0) when the system reset switch is depressed, and remains low until the 8155 chip is initialized. Initialization of the 8155 resets the jump-start latch. To properly perform this function, the port C of the 8155 must be initialized to an output port first, even though it will then be changed to an input port. Any further excursions of the port C pin will not change the reset condition of the latch.

W10

This jumper is installed. This jumper connects to pin 68 of the bus, and provides the M-write function. If this function is provided by another board in the system, remove W10 from the CPU card.



W11

This jumper is installed. W11 conflicts with W12 and only one can be installed. W11 causes the SMEMR pin on the bus to be controlled by the PDBIN signal. This function is required in some systems.

W12

This jumper is not installed. W12 conflicts with W11 and only one can be installed. W12 provides the SMEMR status function from the decoder status output of the 8085 CPU chip.

SERIAL I/O STRAP OPTIONS

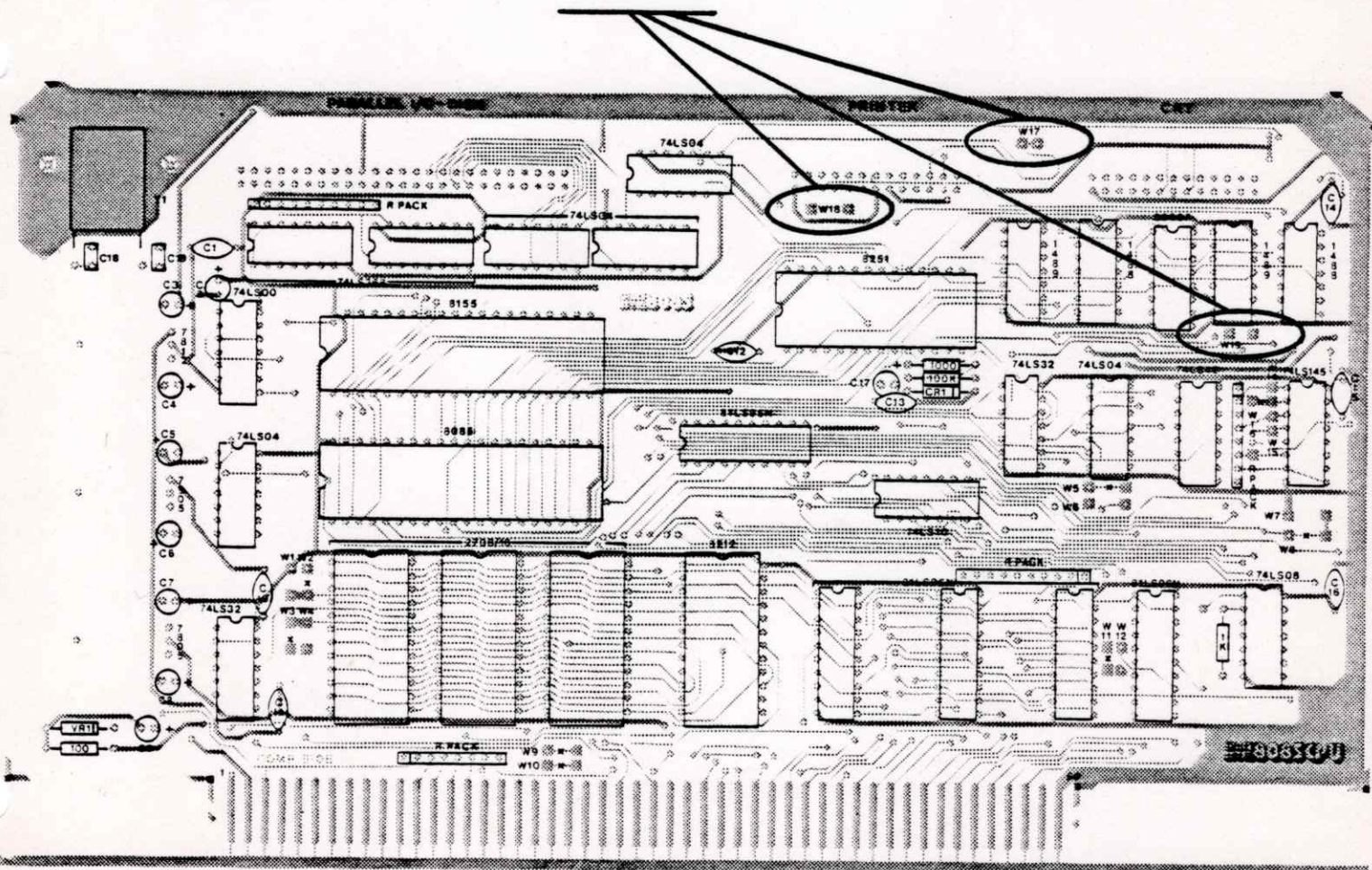
W17 This jumper is installed. The SID pin of the CPU chip is buffered by an EIA RS232C receiver, and parallels the CRT keyboard EIA RS232C input. This is necessary to allow the SPACE BYTE 8085 PROM monitor to automatically determine the correct CRT speed. The jumper may be removed for single board controller applications, thus providing one additional input, and one additional output pin to the system.

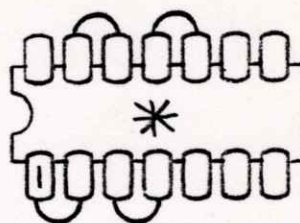
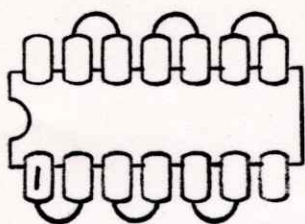
W18

This jumper is not installed. This jumper turns on the CTS function for the printer port. Install this jumper if the printer does not provide the function.

W19

This number is not installed. This jumper connects the RTS and CTS functions together on the CPU card. This jumper may be required if your CRT does not provide compatible handshaking. These functions appear on pins 4 and 5 of the EIA data connector. This jumper must be installed when using the ADM-3 or SOROC IQ 120.

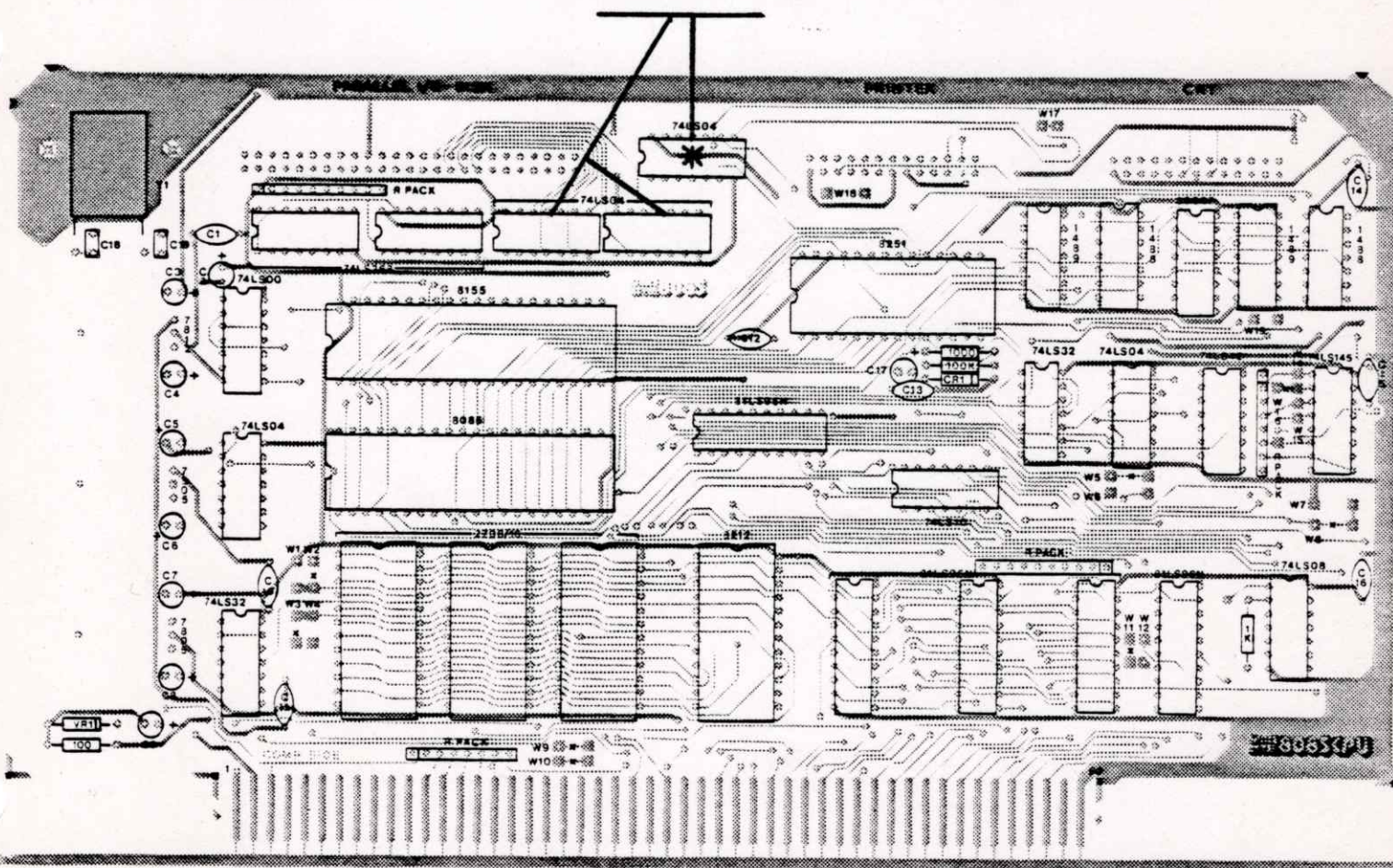




PARALLEL BUFFERED OUTPUTS

16 lines of buffered I/O are provided and configured for the ICOM Disk system. This buffering may be removed, and jumper plugs inserted, or other buffer chip types may be used instead of the 74LS04 chips that are provided. Other chips that would increase drive capability are the 7404, 74S04, and open collector versions, 7406, 7407. As shipped the 8085 CPU card will directly interface to the ICOM Disk system, using the FC-360 disk controller supplied by ICOM.

When removing all buffer chips and substituting direct jumpers, the ports may be used as input as well as outputs. Please note that the chip marked with the star must be left installed, with pins 1 and 3, and pins 13 and 11 out of the socket, and jumpers made to the back of the card as shown. This is because there are two inverter stages in the chip used for USART control and may not be removed.



INSTALLATION INSTRUCTIONS POLYMORPHIC VIDEO DRIVER CARD

The PROM Monitor Software included with your SPACE BYTE 8085 CPU card will support a video card as the console device.

There are several modifications necessary to the card before it will operate with the 8085 CPU. The primary change involves the divider chain formed by IC1 and IC2, that divides the clock reference to a frequency of 15,750 Hertz needed for faster scan generation.

The 8085 CPU supports pin 24 of the edge connector (phase 2 clock) but the frequency present at this pin is 3.072 MHz instead of the more common 2.0 MHz found in 8080 systems. Modification of the divider chain is necessary to provide the frequency closest to 15,750 Hertz at its output. The frequency produced is not exactly 15,750 Hertz so some small adjustment in the horizontal hold control on the video monitor may be necessary.

The 8085 CPU card does not support pin 49 of the edge connector. This is normally phase 2 inverted.

Details of steps necessary to modify the video card

- 1) Add a wire jumper between edge connector pins 24 and 49.
- 2) Remove any wires or etched jumpers from the 3 holes marked JMP1 (near IC1 and IC2).
- 3) Cut etch connecting pins 4 and 7 of IC2.
- 4) Add a wire connecting together pins 3, 4, and 5 of IC2.
- 5) Remove etch from pins of IC1, isolating pins 1, 3, 4, 5 and 7 from each other and from any other pins.
- 6) Add a wire connecting together pins 5 and 6 of IC1.
- 7) Set the board address switch for F800H.

Install the video card in the mainframe and connect it to the keyboard and video monitor.

NOTE: No RAM need be installed for this test.

Turn on the system and depress 'A' on the keyboard. The display will most likely have a tear in the horizontal. Adjust the horizontal hold control of the display monitor until a stable display is present.

Depress R(CR) on the keyboard. This will cause a complete line of characters to be displayed. Adjust the two potentiometers on the video card for line length and starting character position, until the line is centered on the display monitor. Using the 'D' command, display 0 to 100H. This will fill the screen with characters, and point up the need for any other adjustments to the monitor.

Operate the system according to the SPACE BYTE PROM Monitor operating instructions.

OPERATING INSTRUCTIONS

SPACE BYTE 8085 PROM MONITOR

The SPACE BYTE 8085 CPU is delivered complete with an extended monitor programmed into 3 TMS2708JL EPROMS. This monitor program executes at address C000H and extends to address CBFFH. The monitor supports the I/O devices and contains a software debugging capability that allows assembly language programs to be executed to software breakpoints that return control to the monitor.

In addition, the following support software has been included to simplify the initial use of the CPU.

1. Serial EIA CRT I/O routines.
2. Serial EIA Printer I/O routines.
3. ICOM Disk I/O routines.
4. Tarbell tape read and write to cassette.
5. SPACE BYTE 2708/2716 EPROM Programmer controls.
6. Polymorphic video driver routines.

The SPACE BYTE 8085 CPU fully supports the CRT, Serial Printer, and the disk interface, however, the Tarbell and Polymorphic software require additional hardware at the system level. Cable assemblies complete the connection to the rear panel of the system where 25 pin female data sockets provide easy connection to standard CRT and printer units.

The monitor software supports the indicated devices, and because all related RAM requirements are met by the on-card scratch RAM located at address D800H, the main RAM memory is usable from location 000H thru BFFFH, allowing for 48K RAM systems.

MONITOR COMMANDS

Monitor commands consist of a single alpha character typed on the console device keyboard, followed by parameters describing the complete job to be accomplished.

To provide minimum problems in getting this product to "run" in your existing system, the monitor provides for auto-device-speed selection. The software will determine automatically, upon restart, if a Polymorphic Video Driver card is installed in the mainframe, and/or if a CRT device is connected to the serial EIA port provided on the 8085 CPU card. Additionally, the CPU automatically sets the baud rate on the CPU to match that of the CRT.

The user simply strikes the letter "A" on the keyboard and the system then automatically assigns that device as the programmers console.

If a CRT is connected the 8085 CPU will automatically set the speed providing it is one of the following:

110
150
300
1200
1800
2400
4800
9600

No speed restrictions are placed on the Polymorphic system video driver. The apparent speed is approximately 20,000 (2000 characters per second) and is set by processor loop delays in the driver software.

When the letter "A" is depressed, the system will reply with the system greeting and the top of RAM calculation.

SPACE BYTE 8085 Monitor V-1.0 top of RAM at XXXX

The RAM installed in the computer will be sampled starting at address 0000H and the 4 digit number indicated at the end of the greeting message is the highest memory address found in contiguous RAM. If no RAM is installed in the mainframe, the number will be 0000.

Two software stacks are used in the 8085 CPU. The monitor stack is located at D8FFH. The user stack, assigned whenever the "G" command is invokled, instantly transfers the CPU to the user stack. The user stack is automatically located at the top of contiguous RAM memory.

TARBELL MAGNETIC TAPE CASSETTE PROGRAMS

Four commands are dedicated to controlling the Tarbell Cassette interface board, if installed in the mainframe. If the interface board is not present in the system, use of the commands is prevented, and the reply message:

NO MAG TAPE

is displayed on the console device.

PROM PROGRAMMING ROUTINES

If the optional SPACE BYTE 2708/2716 EPROM Programmer is connected to the 8085 CPU card, it is possible to program any 2708 device, and the TMS2716 TI device. Three monitor commands control the operation of the programmer.

SYSTEM MONITOR COMMAND TABLE

After Initialization

"A" Assign	Assigns list device (CRT or Printer)
"B" Boot	Loads disk executive
"C" Compare	Compare PROM to RAM
"D" Dump	Display memory contents
"E" Tape-Execute	Load and execute program from Tarbell Cassette
"F" Fill	Fill memory with constant
"G" Go	Go to RAM location XXXX (with break points)
"H" Hexarith	Hexidecimal arithmetic function
"I" Inspect	Inspect and change memory
"J" Jump	Jump to location 0000
"L" Tape-Load	Load only from Tarbell Cassette
"M" Move	Move a block of memory
"N" Null	Set nulls to console device
"P" PROM	Program PROM
"R" Registers	Examine and change CPU registers
"S" Speed	Set baud rate to Printer
"T" Transfer	Transfer PROM to RAM
"V" Verify	Check sum verification of mag tape
"W" Write	Write file to Tarbell Cassette

A ASSIGN

The assign command determines the destination of data handled by the printer I/O routine. When the CPU has power applied initially or when the reset switch is depressed, this data is routed to the console device.

When the letter "A" is struck, the system replies with the prompt message: A list device=

There are only 2 proper assignment characters. "C" assigns the console device, and "P" assigns the printer. The result will appear as a complete message:

A LIST DEVICE = CRT

or

A LIST DEVICE = PRINTER

Any other character entered will error, displaying the error message:

KEYBOARD ERROR! (Bell)

and leaving the assignment unchanged.

B (CR) BOOT DISK EXECUTIVE

This command jumps to the address of the program that loads the disk executive from disk into RAM. If no disk system is connected or if power to the disk system has been removed, the reply message:

NO DISK

will be displayed on the console device.

If the disk system is connected and power is applied, but the diskette is missing or not up to speed, the reply message:

DISK NOT READY

will be displayed on the console device.

If the disk system is connected, a diskette containing the disk executive as the first file must be inserted into drive 0.

The bootstrap loader will then load the disk executive and jump to location 0, when the program is loaded. The result will be the greeting:

ICOM FDOS-III SPACE BYTE VER 1.0

The disk system is now operational. Refer to the operation section of the disk system manual for the available disk commands.

Several types of problems may occur that will prevent loading of the disk executive. If the installed diskette does not have the executive object code as the first file, the first file encountered will be loaded anyway. The loader has a size restriction and if the file that is being loaded is larger, the system will display:

FILE SIZE ERROR

and control will be returned to the PROM monitor. If the first file has any errors recorded, the message:

CHECKSUM ERROR

will be displayed, and control will be returned to the system monitor.

NOTE: The PROM monitor uses a single prompt character (>) and the disk system executive program uses an exclamation mark (!)

If the first file is object, and if it is smaller than the size restriction, and if it happens to be assembled to execute at location 0, it will be loaded and executed. No error messages will be displayed, because no detectable errors have been made. It is possible to have no indication appearing on the console device that any program is running.

Reset the system by depressing the reset switch, and depressing "A" on the console keyboard.

C *,*(CR)

COMPARE PROM TO RAM

(*=HEX NUMBER)

The compare command is useful in determining the condition of a PROM. If the contents of the PROM compares to a block of FFH characters, the PROM is totally erased and ready for programming. If the PROM has just been programmed, it can be compared against the area of RAM used to program it to determine if the programming operation was successful. The command:

C2000,23FF (CR)

will compare the contents of a 2708 device to the contents of RAM memory locations 2000H to 23FFH. If the compare operation is completely successful the reply message:

DONE

will be displayed on the console device. If the PROM does not compare with the specified area of RAM memory, the error message:

DOES NOT COMPARE

will be displayed on the console device. The compare command can also compare the contents of the PROM against areas of PROM memory in the system, such as the PROM monitor itself.

D *,*(CR)

DISPLAY MEMORY

The "D" command for "Display" or "Dump" uses the printer I/O routine. If the printer is assigned as the list device, the area of memory specified will be printed, otherwise it will be displayed on the console device.

After typing "D", the number (one to four digits) representing the low end or start is entered, followed by a comma, and then the high end or stop address, followed by a return.

Number entered must be in hexadecimal notation. Leading zeros need not be entered.

The display format includes the current memory address at the start of each display line, followed by 16 bytes of data in hex notation.

E *,*(CR)

EXECUTE MAGNETIC TAPE FILE

This command is the same as the "L" command except that control is transferred to the program that has just been loaded, instead of to the PROM monitor. (use this command when initializing Basic)

F *,*,*(CR)

FILL MEMORY

The fill command is used to load areas of RAM memory with a constant. The entry format is:

F (START ADDRESS), (STOP ADDRESS), (CONSTANT) (CR)

All numerical entries are in hexadecimal notation. The following entry will fill RAM memory between 0100H and 01FFH with zeros.

F100,1FF,0(CR)

One valuable use for the fill command is to set a block of memory to FFH when using the compare command. The compare command, compares the contents of a PROM inserted into the optional SPACE BYTE PROM Programmer, with an area of memory. A totally erased PROM contains all bits high, or a block of FFH. This test quickly indicates when a PROM is erased and ready for programming.

G (*) (SPACE BAR*) (SPACE BAR*) (CR) DEBUGGING ROUTINES

The go command controls the execute (go to location) and breakpoint routine that provides some of the debugging capability of the SPACE BYTE 8085 CPU. The other command is the register (R) command that will be described separately. The go command can take a number of optional parameters. Variations are:

1. G1000(CR)
2. G1000(SPACE BAR)1100(CR)
3. G100(SPACE BAR)1100(SPACE BAR)1200(CR)
4. G(SPACE BAR)1100(CR)
5. G(SPACE BAR)1100(SPACE BAR)1200(CR)
6. G(CR)

The first example executes the program at RAM location 1000H. No breakpoints are set and no automatic return to the PROM monitor is possible. Before jumping to location 1000H, the CPU registers are loaded with the values present in the user save area, and the user stack pointer (located at the top of RAM) is set. The address used with the "G" command in this instance, can be any address in memory, since no breakpoints have been set.

Example 2 includes one breakpoint set at address 1100H.

NOTE: When a breakpoint is set, address locations 0 thru 3H are used by the breakpoint routine and must not appear in the go command.

The breakpoint system functions by first saving the program code at locations 0 thru 3 of RAM memory. A jump command is then written at location 0. At the address specified for the breakpoint (in this case 1100H), the byte of user code is saved, and the instruction RST0 (C7H) is substituted. When this location is encountered during program execution it causes a call (one byte) to location 0 to be performed. The jump instruction at location 0, vectors the CPU back to the monitor to a point where all CPU registers contents will be saved before monitor execution is allowed. At the same time, user code is restored to all modified parts of the user program including the jump address at location 0. The contents of all CPU registers are then displayed on the console device in the following format:

A = ACCUMULATOR
 B = B REGISTER
 C = C REGISTER
 D = D REGISTER
 E = E REGISTER
 F = CPU FLAGS
 I = INTERRUPT MASK REGISTER
 H = H REGISTER
 L = L REGISTER
 M = MEMORY POINTER (SAME AS H-L)
 P = PROGRAM COUNTER (ONE MORE THAN BREAKPOINT LOCATION)
 S = STACK POINTER (USER STACK)

At the programmers option, individual register contents can now be changed before program debugging is resumed.

Once a breakpoint is encountered, the location of the instruction is saved, and if program execution is desired at that point, the address need not be specified in the next go command.

For example: G(CR) would cause execution to resume at location 1100H, and of course would set no breakpoints.

Example 3 sets two breakpoints at the same time. This is useful if the program can take one of two paths at the point of interest. Encountering either breakpoint will return control to the monitor. The "P" register (program counter) will display one more than the address of the breakpoint encountered. A maximum of two breakpoints can be set.

Examples 4, 5, and 6 are formats that can be used only after a breakpoint has been encountered during the debugging session.

H *1,*2(CR)

HEXIDECIMAL ARITHMETIC

This feature provides a hexadecimal adder/subtractor for the convenience of the programmer. When two numbers are entered, the sum and difference of the two numbers are then displayed on the console device.

The sum of the two numbers is displayed first, followed by the result of subtracting number 2 from number 1.

I *(SPACE BAR)

INSPECT MEMORY

The inspect memory command allows single or multiple bytes of memory to be inspected (or viewed) and in the case of RAM memory, to change the contents of selected locations.

Entering:

I(ADDRESS) (SPACE BAR)

will display the data at that address on the console device. Depressing the space bar again will advance to the next byte, and display it without changing the previously viewed data. Inspecting the data at address 1000H would look like this after depressing the space bar:

I1000 15-

The memory location contents (hex 15) can be changed by entering new data via the keyboard at this time. To enter the data type the new value and then operate the space bar again, or to terminate the inspect command enter a return (CR).

J JUMP

This command executes an unconditional jump to memory location 0. It differs from the go command in that no user stack is created, nor are any of the CPU registers modified before the jump is made. No (CR) terminator is required for this command.

System software such as the assembler, editor, and disk executive programs all start at location 0. The "J" command offers a convenient way of returning to the disk executive from the monitor.

L *,*(CR) LOAD CASSETTE FILE FROM TAPE

The "L" command provides a load routine that will load the file into RAM memory and then return control to the PROM monitor.

The tape file may be loaded to any place in RAM, but to be executed there, assuming an object program is being loaded, the exact starting and ending addresses must be known beforehand. These addresses are now recorded on the tape. The tape record consists of a starting sync character followed by the tape file in binary, followed by a single checksum byte. In order for the checksum system to work, the ending address entry minus the starting address entry must equal the byte count minus the sync character and the recorded checksum character.

The correct technique is to type in the entry except for the terminator (CR), start the tape cassette, then depress the return key. The cursor will return to the start of the current line and remain there until the specified number of bytes have been loaded. Control will then be returned to the 8085 PROM monitor, and no error will have occurred. If a load error has occurred the message:

CHECKSUM ERROR

will be displayed on the console device if the calculated checksum did not match the recorded checksum or

* MEMORY WRITE ERRORS

will be displayed if the checksum matched but errors were detected in writing the file to RAM memory. This is a type of memory test. Each byte from tape is written to RAM and then RAM is read and compared against what was written. If a mismatch occurs, the problem can only be a defective RAM location. The number of such defective reads are displayed at the start of the error message. Since an error of this type is rare, the maximum number of errors that can be displayed, is 255. For normal operation this type of error should not occur.

M *,*,*(CR) MOVE CONTENTS OF MEMORY

The contents are really copied into the new locations, but not removed from the old locations. This command can be used to transfer any size block resident in RAM or PROM into a new area that must be located in RAM memory. The command format is:

M(OLD START ADDR), (OLD ENDING ADDR), (NEW START ADDR) (CR)
M,7000,7FFF,0

N *(CR) NULL

The null command adds a delay in the form of nulls (up to 9) to the CRLF routine used by the console device. The delay is set to 0 when the CPU has power initially applied, or whenever the reset switch is depressed. Most CRT terminals do not require delays during CRLF (Carriage Return Line Feed) operations, but printing terminals that may be used as console devices sometimes require a rather long delay to allow the print head carriage to reach the left-hand margin before the first character of the next line in output.

Some older CRT terminals also require one or two nulls during CRLF operations especially at the higher baud rates.

The CRLF routine used by the printer software includes 7 nulls as standard, and they are not removable.

P *,*(CR) PROGRAM A PROM

This command copies (programs) the area of RAM memory (or PROM memory) specified in the command into the PROM inserted into the PROM programmer socket. The area specified must be exactly 1024 bytes for a 2708 device, and 2048 bytes for a 2716 device.

NOTE: The PROM programmer can also program 2704 devices.
specify areas of RAM 512 bytes in length for all
commands.

Programming a 2708 device takes 3.5 minutes, a 2716 device takes
7.0 minutes. When finished the reply message:

DONE

will be displayed on the console device. Be sure to do a compare
command against the same area of RAM to verify the correctness of
the programming operation.

R REGISTER(A) (B) (C) (D) (E) (F) (I) (H) (L) (M) (P) (S) (CR)

The register command and its many optional parameters are used to
modify the CPU registers during a debugging session.

R(CR)

will display the contents of all CPU registers on the console
device. Keep in mind that these are not the absolute current
values, but the values that will be present after the go command
has been executed.

During initial application of power, or if the reset switch is
depressed, or if a control-C command is entered while in the
monitor, these registers are set to 0, except for the stack
pointer, which is set to the top of user RAM.

During program execution, the display of the registers will
reflect their states at the time the program encountered the
breakpoint address. To change the values in these registers,
type R(*) (CR), where * equals one of the following letters:

A
B
C
D
E
F
I
H
L
M
P
S

The display will indicate the present contents followed by a dash.
New values for the register can now be entered from the keyboard.
Depressing the (CR) key will terminate the command, depressing
the space bar will display the contents of the next register for
possible changes.

S (BAUD RATE) (CR) SPEED

The speed command sets the baud rate for the serial printer attached to the 8085 CPU. The following rates are available:

110
150
300
1200
1800
2400
4800
9600

To set the baud rate to 1200 for example, enter:

S1200(CR)

This command does not change the baud rate assigned to the console device. If the entry is made incorrectly, the keyboard error message will be displayed and the bell will sound. If the entry is very close to one of the listed rates, the software will default to the nearest rate without displaying the error message.

T *,*(CR) TRANSFER PROM TO RAM

The transfer command copies the contents of the PROM into the specified area of RAM memory. The command entry format is:

T(STARTING ADDR),(ENDING ADDR)(CR)

For a 2708 device the ending address minus the starting address should be exactly 1024 bytes. (400H) for the 2716 device the area of RAM needed is 2048 bytes, (800H).

The command:

T2000,23FF(CR)

will transfer the contents of a 2708 device to RAM memory locations 2000H thru 23FFH. One of the uses for the command is to read PROM contents into RAM for patching and programming another PROM.

The command:

T2000,27FF(CR)

will transfer the contents of a 2716 device to RAM memory locations 2000H thru 27FFH.

NOTE: Care should be exercised when inserting the PROMS into the zero insertion force sockets on the programmer. One socket is wired for the 2708 device, and the other is wired for the 2716 device. Only one PROM can be inserted at a time. All PROMS should be removed for operation of the disk system.

V (*,*)(CR) TAPE VERIFY

Using the verify command (you must rewind the tape), allows comparing the tape content to the contents of RAM, without changing

either in the process. Command entry is the same format as for load, and if successful, the reply message:

TAPE VERIFIED

will be displayed on the console device.

If any bytes do not compare, the error message:

CHECKSUM ERROR

will be displayed on the console device.

W *,*(CR)

WRITE MAGNETIC CASSETTE TAPE FILE

The command entry format is the same as for all mag tape commands:

W(START ADDRESS),(ENDING ADDRESS)(CR)

The correct technique is to type the entry except for the terminating (CR), start the tape cassette (watch for the tape to pass the head), then depress the return key. When the file has been completely written to tape, control is returned to the monitor. Its a good idea to rewind the tape, and verify correct recording by using the tape verify command.

SPACE BYTE 8085 CPU

Parts List

11-28-77

ITEM	REQ.	PART #	DESCRIPTION	SUPPLIER
1	1	-	Printed Circuit Board	
SEMICONDUCTORS				
2	1	8085	Central Processor	Intel
3	1	8155	Parallel Port	Intel
4	1	8251	USART	Intel
5	1	8212	8-Bit Latch Tri-State	Intel
6	5	74LS04	Hex Inverter	Texas Instruments
7	1	340097	Hex Buffer Tri-State	National
8	2	74LS32	Quad 2-Input or Gate	Texas Instruments
9	2	1488	RS232 Driver	Texas Instruments
10	2	1489	RS232 Receiver	Texas Instruments
11	1	81LS96	Octal Tri-State Buffer	National
12	4	81LS95	Octal Tri-State Buffer	National
13	3	2708	1024 x 8 EPROM	Texas Instruments
14	1	74LS00	Quad 2-Input NAND Gate	Texas Instruments
15	1	74LS08	Quad 2-Input AND Gate	"
16	1	74LS10	Triple 3-Input NAND Gate	"
17	1	74LS42	4 Line to 10 Line Decoder	"
18	2	74LS368	Hex Inverting Tri-State Buffers	"
19	1	74LS145	4-Line to 10-Line Decoder	"
20	1	7805	5 Volt Positive Regulator	"
21	1	7812	12 Volt Positive Regulator	"
22	1	7905	5 Volt Negative Regulator	"
23	1	IN4742	12 Volt Zener Diode	Siemens
24	1	IN4148	Signal Diode	Siemens
MISC. PARTS				
25	3	609-502M	Heat Sinks	Thermalloy
26	1	609-5002M	2 x 25 Right Angle Header	Ansley
27	2	609-2602M	2 x 13 Right Angle Header	Ansley
28	2	C834002	40 Pin Dip Sockets	Texas Instruments
29	1	C832802	28 Pin Dip Sockets	"
30	4	C832402	24 Pin Dip Sockets	"
31	5	C832002	20 Pin Dip Sockets	"
32	5	C831602	16 Pin Dip Sockets	"
33	14	C831402	14 Pin Dip Sockets	"
34	1		6.144 MHz Crystal	
35	8	UK16-184	.1 @ 16V Disc Capacitor	
36	9	TAG10-35	10 @ 35V Tantalum Capacitor	
37	2	CM04ED200J03	20 PF Dipped Mica Capacitors	ITT
38	2		1K Resistor 1/4 W 5%	
39	2	4308-101-102	8 Resistor 1K Sip Packs	Bourns
40	2	4310-101-102	10 Resistor 1K Sip Pack	Bourns
41	1		100K Resistor 1/4 W 5%	
42	1		100 OHM 1/4 Watt Resistor	
43	2	1G-A35-12	Cable Assemblies, 25 Pin Female Data to 26 Pin Wire Wrap Post Socket. 12 Inches Long.	

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1 0000 TITLE 'SPACE BYTE 8085 FROM MONITOR 11-14-77'
2 0000
3 0000 ; THE SPACE BYTE CORPORATION
4 0000 ; 1720 PONTIUS AVE.
5 0000 ; SUITE 201
6 0000 ; LOS ANGELES CA. 90025
7 0000 ; (213) 488-8080
8 0000

```

9 0000 ;EQUATES LIST

```

10 0000
11 0000 00D3 PORTI EQU 0D3H ;8155 STATUS CONTROL REG.
12 0000 00D9 PORTA EQU 0D9H ;8155 'A' 8-BIT PORT
13 0000 00DA PORTB EQU 0DAH ;8155 'B' 8-BIT PORT
14 0000 00DB PORTC EQU 0DBH ;8155 'C' 8-BIT PORT
15 0000 00DC PORTD EQU 0DCH ;DISK DATA INPUT PORT
16 0000 00DD TIMLO EQU 0DDH ;8155 TIMER PORT (LO BYTE)
17 0000 00DD TIMHI EQU 0DDH ;8155 TIMER PORT (HI BYTE)
18 0000 00C0 SDATA EQU 0C0H ;USART (8251) DATA PORT
19 0000 00C1 SSTAT EQU 0C1H ;USART STATUS/CONTROL
20 0000 0001 TXRDY EQU 1 ;USART TX BUFFER EMPTY.
21 0000 0002 RXRDY EQU 2 ;USART RX BUFFER FULL.
22 0000 006E CASC EQU 6EH ;MAG TAPE CONTROL PORT.
23 0000 006F CASD EQU 6FH ;MAG TAPE DATA PORT.
24 0000 4000 MODE EQU 4000H ;OFFSET FOR CONSTANTS
25 0000 4014 S9600 EQU MODE+20 ;9600 X 16
26 0000 4028 S4800 EQU MODE+40 ;4800 X 16
27 0000 4050 S2400 EQU MODE+80 ;2400 X 16
28 0000 406B S1800 EQU MODE+107 ;1800 X 16
29 0000 40A0 S1200 EQU MODE+160 ;1200 X 16
30 0000 42B0 S0300 EQU MODE+640 ;300 X 16
31 0000 4500 S0150 EQU MODE+1280 ;150 X 16
32 0000 46D1 S0110 EQU MODE+1745 ;110 X 16
33 0000 F800 PAGE EQU 0F800H
34 0000 00F8 PSTAD EQU 0F8H
35 0000 0024 READP EQU 24H ;FROM PROGRAMMER READ CODE.
36 0000 0010 PPULS EQU 10H ;PROGRAMMER 24V PULSE BIT
37 0000 000E WRTP EQU 0EH ;PROGRAMMER WRITE CODE
38 0000 0001 STB EQU 1 ;PROGRAMMER DATA STROBE
39 0000 ;
40 0000 RIM MACRO
41 0000 DB 20H
42 0000 ENDM
43 0000 ;
44 0000 SIM MACRO
45 0000 DB 30H
46 0000 ENDM
47 0000
48 0000
49 0000 ;SCRATCHPAD RAM REGISTER ASSIGNMENTS
50 0000
51 0000 D800 ORG 0D800H
52 D800
53 D800 D8FF STACK EQU $+255
54 D800 NULOC: DB 1 ;CONSTANT FOR NULLS IN ORLF
55 D801 HIBYTE: DB 2 ;TOP OF MEMORY STORED HERE.

```


SPACE BYTE 3085 FROM MONITOR 11-14-77

56 D803	PSTAT:	DS	1	;PRINTER SELECT STATUS
57 D804	BAUDP:	DS	2	;SPEED SELECTED FOR PRINTER
58 D806	ERBYT:	DS	1	;ERROR DURING LOAD
59 D807	CSTAT:	DS	1	;CONSOLE DEVICE STATUS
60 D808	PASS:	DS	1	
61 D809	OFILE:	DS	1	;OUTPUT FILE
62 D80A	OUNIT:	DS	1	;OUTPUT DRIVE NUMBER
63 D80B	IUNIT:	DS	1	;INPUT DRIVE NUMBER
64 D80C	ISIZE:	DS	2	;INPUT FILE SIZE IN HEX SECTORS
65 D80E	ITRK:	DS	1	;INPUT FILE TRACK NUMBER
66 D80F	ISCTR:	DS	1	;INPUT FILE SECTOR NUMBER
67 D810	ICNTR:	DS	1	;INPUT FILE CHARACTER COUNT
68 D811	OSIZE:	DS	2	;OUTPUT FILE SIZE IN HEX SECTORS
69 D813	OTRK:	DS	1	;OUTPUT FILE TRACK NUMBER
70 D814	OSCTR:	DS	1	;OUTPUT FILE SECTOR NUMBER
71 D815	OCNTR:	DS	1	;OUTPUT FILE CHARACTER COUNT
72 D816	TITRK:	DS	1	;INPUT FILE TRACK NUMBER (RESTORE)
73 D817	TISZE:	DS	2	;INPUT FILE SIZE (RESTORE)
74 D819	BAUDC:	DS	2	;SPEED SELECTED FOR CRT
75 D81B	ALOC:	DS	1	;ACCUMULATOR STORAGE
76 D81C	BLDC:	DS	1	;REG. B STORAGE
77 D81D	CLDC:	DS	1	;REG. C STORAGE
78 D81E	DLDC:	DS	1	;REG. D STORAGE
79 D81F	ELDC:	DS	1	;REG. E STORAGE
80 D820	FLDC:	DS	1	;FLAG REGISTER STORAGE
81 D821	ILDC:	DS	1	;INTERRUPT STORAGE
82 D822	HLDC:	DS	1	;REG. H-L STORAGE
83 D823	LLDC:	DS	1	;REG. L STORAGE
84 D824	D822	MLDC	EQU	HLDC
85 D824	PLDC:	DS	2	;PROGRAM COUNTER STORAGE
86 D826	SLDC:	DS	2	;STACK POINTER STORAGE
87 D828	TLDC:	DS	12	;TRAP INFORMATION STORAGE
88 D834	EXIT:	DS	2	;USER ADDRESS STORE
89 D836	CURS:	DS	2	
90 D838	IFL:	DS	1	
91 D839	WFL:	DS	1	
92 D83A	UND:	DS	1	
93 D83B	SPD:	DS	1	
94 D83C	TEMP:	DS	2	;TEMPORARY STORAGE.
95 D83E				
96 D83E				
97 D83E	C000	ORG	0C000H	
98 C000				
99 C000	C357C0	START:	JMP	INIT ;JUMP AROUND VECTOR TABLE.
100 C003				
101 C003				;SUBROUTINE VECTOR TABLE
102 C003				
103 C003	C387C3	INCV:	JMP	INC ;CONSOLE INPUT
104 C006	C3C9C9	READV:	JMP	CASIN ;MAG TAPE INPUT
105 C009	C3B1C2	OUTCV:	JMP	OUTC ;CONSOLE OUTPUT
106 C00C	C365C9	TWRTV:	JMP	CASOT ;MAG TAPE OUTPUT
107 C00F	C3D4C9	RIV:	JMP	RI ;READ A BYTE FROM DISK
108 C012	C370C9	WRTV:	JMP	WRT ;WRITE A BYTE TO DISK
109 C015	C393C5	OUTPV:	JMP	OUTP ;PRINTER OUTPUT
110 C018	C3E1C3	MEMSZ:	JMP	MEMTP ;COMPUTE MEMORY SIZE

SPACE BYTE 8085 PROM MONITOR 11-14-77

```

111 001B C30DC9  RESTV:  JMP      RESET      ;RESET ERRORS AND HOME HEAD.
112 001E C3BCCA  KUSV:   JMP      KUS        ;TRANSMIT UNIT SECTOR BYTE
113 0021 C3CBCA  XKUSV:  JMP      XKUS       ;
114 0024 C3D3CA  SEEKV:  JMP      SEEK+1    ;JUMP TO SEEK TRACK
115 0027 C3DFCA  FLAGV:  JMP      RFLAG     ;RESET DISK ERROR FLAGS
116 002A C3E1CA  LOOPV:  JMP      LOOP      ;COMMAND AND LOOP WHILE BUSY
117 002D C333C9  RSTRV:  JMP      RSTR      ;RESTORE INPUT FILE POINTERS
118 0030 C315CB  PASSV:  JMP      IPASS     ;INTERPASS FUNCTIONS
119 0033 C399C2  CRLFV:  JMP      CRLF      ;
120 0036          ;
121 0036 CD4CC9  ASSEM:  CALL     LOADER
122 0039 CD33C9          CALL     RSTR
123 003C C33DD8          JMP      0D3DDH
124 003F          ;
125 003F CD4CC9  EDITOR: CALL     LOADER
126 0042 CD33C9          CALL     RSTR
127 0045 C34000          JMP      40H
128 0048          ;
129 0048 31FFD8  UPDAT:  LXI      SP,STACK
130 004B CD17C9          CALL     BOOT1
131 004E C34300          JMP      43H
132 0051          ;
133 0051 CD4CC9  PROG:   CALL     LOADER
134 0054 C32EC1          JMP      MAIN
135 0057          ;
136 0057          ;INIT...INITIALIZE HARDWARE AND SOFTWARE REGISTERS.
137 0057          ;
138 0057 31FFD8  INIT:   LXI      SP,STACK      ;SET MONITOR STACK
139 005A 97          SUB      A              ;CLEAR A
140 005B 3200D8          STA      NULLC        ;SET NULLS TO 0.
141 005E 3203D8          STA      PSTAT      ;SELECT CRT = LIST DEVICE
142 0061 3238D8          STA      IFL        ;SET VIDEO TO NEED RESET
143 0064 3ECF          MVI      A,0CFH
144 0066 D3D8          OUT      PORTI
145 0068 CDF2C7          CALL     VIDEO      ;INIT VIDEO BOARD
146 006B          ;
147 006B          ;USART PORTS (3251)
148 006B          ;
149 006B 3EFA          MVI      A,0FAH      ;SET USART FOR 7 BITS
150 006D D3C1          OUT      SSTAT      ;EVEN PARITY, 2 STOP BITS.
151 006F 3E25          MVI      A,25H
152 0071 D3C1          OUT      SSTAT
153 0073          ;
154 0073          ;INIT PARALLEL PORTS
155 0073          ;
156 0073 211440          LXI      H,39600    ;SET PRINTER SPEED
157 0076 2204D8          SHLD     BAUDP      ;TO 9600 BPS.
158 0079 DBF9          INIT1: IN       0F9H      ;KEYBOARD PORT STATUS
159 007B FEFF          CPI      -1          ;FF = NO BOARD INSTALLED
160 007D CA9CC0          JZ       INITB      ;GOTO CRT AND LOOK
161 0080 E601          ANI      1          ;VIDEO BOARD INSTALLED
162 0082 CA9CC0          JNZ      INITB      ;NO KEY DEPRESSED
163 0085 DBF8          IN       0F8H      ;KEYBOARD PORT DATA
164 0087 FE41          CPI      41H        ;WAS AN A DEPRESSED?
165 0089 CA11C1          JZ       INITD      ;YES...

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

166 C08C      +INITB: RIM      ;WAIT
      C08C 20      +      DB      20H
      C08D
167 C08D 17      RAL          ;FOR START
168 C08E DA79C0   JC          INIT1 ;BIT
169 C091 210000   LXI          H,0
170 C094 E5      PUSH         H
171 C095      +INIT2: RIM      ;THIS
      C095 20      +      DB      20H
      C096
172 C096 E1      POP          H      ;LOOP TAKES
173 C097 23      INX           H      ;14.9739
174 C098 E5      PUSH         H      ;MICROSECONDS
175 C099 17      RAL          ;TO
176 C09A D295C0   JNC          INIT2 ;COMPLETE
177 C09D E1      POP          H
178 C09E      ;
179 C09E      ;NUMBER IN H-L RELATES TO WIDTH OF START PULSE
180 C09E      ;
181 C09E AF      XRA          A      ;CLEAR A
182 C09F 3207D8   STA          CSTAT ;SET CRT = CONSOLE
183 C0A2 7C      MOV          A,H
184 C0A3 A7      ANA          A
185 C0A4 CBDDC0   JZ          INIT3
186 C0A7 FE01     CPI          1
187 C0A9 C2B2C0   JNZ          INIT9
188 C0AC 210045   LXI          H,30150 ;H-L = 445
189 C0AF C3CAC0   JMP          INIT4
190 C0B2 FE02     CPI          2
191 C0B4 C279C0   JNZ          INIT1 ;GOODFUP!
192 C0B7 21D146   LXI          H,30110 ;H-L = 507
193 C0BA C3CAC0   JMP          INIT4
194 C0BD 7D      INIT3: MOV          A,L
195 C0BE A7      ANA          A
196 C0BF FA06C1   JM          INITA
197 C0C2 FE0A     CPI          10
198 C0C4 D2DAC0   JNC          INIT5
199 C0C7 211440   LXI          H,39600 ;H-L = 7
200 C0CA 2219D8   INIT4: SHLD         BAUDC ;SAVE IN SCRATCHRAM
201 C0CD 7D      MOV          A,L
202 C0CE D3DC     OUT          TIMLO
203 C0D0 7C      MOV          A,H
204 C0D1 D3DD     OUT          TIMHI
205 C0D3 3ECF     MVI          A,0CFH
206 C0D5 D3DB     OUT          PORTI
207 C0D7 C319C1   JMP          INIT0
208 C0DA FE14     INIT5: CPI          20
209 C0DC D2E5C0   JNC          INIT6
210 C0DF 212340   LXI          H,34800 ;H-L = 14
211 C0E2 C3CAC0   JMP          INIT4
212 C0E5 FE21     INIT6: CPI          33
213 C0E7 D2F0C0   JNC          INIT7
214 C0EA 215040   LXI          H,32400 ;H-L = 28
215 C0ED C3CAC0   JMP          INIT4
216 C0F0 FE2B     INIT7: CPI          43

```


SPACE BYTE 8085 PROM MONITOR 11-14-77

```

217 00F2 D2FBC0      JNC     INIT3
218 00F3 215B40      LXI     H,01800  ;H-L = 37
219 00F8 C3CAC0      JMP     INIT4
220 00FB FE46        INIT3: CPI     70
221 00FD D279C0      JNC     INIT1  ;GOODFUP!
222 0100 21A040      LXI     H,01200  ;H-L = 55
223 0103 C3CAC0      JMP     INIT4
224 0106 FEFF        INIT4: CPI     255
225 0109 D279C0      JNC     INIT1  ;GOODFUP!
226 010B 218042      LXI     H,00300  ;H-L = 223
227 010E C3CAC0      JMP     INIT4
228 0111             ;
229 0111 3EFF        INITD: MVI     A,-1
230 0113 3207D9      STA     CSTAT  ;GET VIDEO = CONSOLE
231 0115 C31CC1      JMP     INITC
232 0119             ;
233 0119             ;INIT STORAGE REGISTER CONTENTS
234 0119             ;
235 0119 CD87C3      INIT0: CALL    INC     ;CLEAN OUT RM REG.
236 011C 2113D9      INITC: LXI     H,ALOC
237 011F 061A        MVI     B,36
238 0121 97          SUB     A          ;CLEAR A
239 0122 77          LOOP2: MOV     M,A
240 0123 23          INX     H          ;INCREMENT MEMORY POINTER
241 0124 05          DCR     B
242 0125 C322C1      JNZ     LOOP2
243 0129             ;
244 0129             ;COMPUTE SIZE OF RAM MEMORY
245 0129             ;
246 0129 210000      LXI     H,0
247 012B CDB0C3      CALL    MILEC
248 012E             ;
249 012E             ;MAIN COMMAND LOOP
250 012E             ;
251 0132 CD4AC5      MAIN:  CALL    CRLF  ;CRLF TO CRT
252 0131 0E3E        MVI     C,'>'  ;PROMPT SYMBOL
253 0133 CDB1C2      CALL    CUTC   ;TO CRT
254 0136 CDB0C1      CALL    ECHO   ;GET KB CHAR AND DISPLAY IT
255 0139 0E02        MVI     C,2    ;GET FOR 2 NUMBERS
256 013B FE03        CPI     3
257 013D CA1CC1      JZ      INITC   ;RESET MONITOR
258 0140 FE41        CPI     'A'
259 0142 CA03C6      JZ      ASSIGN  ;ASSIGN DEVICES
260 0145 FE42        CPI     'B'
261 0147 CAEBC3      JZ      BBOOT   ;LOAD DISK EXEC.
262 014A FE43        CPI     'C'
263 014C CA07C2      JZ      PROM    ;COMPARE PROM TO RAM
264 014F FE44        CPI     'D'
265 0151 C05DC5      CZ      DUMP    ;DISPLAY MEMORY
266 0154 FE45        CPI     'E'
267 0156 CA56C3      JZ      TAPEL   ;LOAD AND EXECUTE FILE FROM MAG TAPE

268 0159 FE46        CPI     'F'
269 015B C039C5      CZ      FILL    ;FILL MEMORY W/CONSTANT
270 015E FE47        CPI     'G'
271 0160 CAEDC6      JZ      GO      ;GO...TO RAM LOCATION XXXX

```

SPACE BYTE 8085 PROM MONITOR 11-14-77

```

373 0163 FE48      CPI      'H'
373 0165 0CE9C3    CZ      HEXAS      ;HEXIDECIMAL ARITHMETIC
374 0168 FE49      CPI      'I'
375 016A 0ADEC5    JZ      INSPECT ;LOOK AT AND CHANGE MEMORY
376 016D FE4A      CPI      'J'
377 016F 0A0000    JZ      0          ;JUMP TO RAM LOCATION 0000
378 0172 FE4C      CPI      'L'
379 0174 0A56C3    JZ      TAPEL      ;LOAD FILE FROM MAG TAPE
380 0177 FE4D      CPI      'M'
381 0179 0C3AC5    CZ      MOVEB      ;MOVE A BLOCK OF MEMORY
382 017C FE4E      CPI      'N'
383 017E 0CA2C1    CZ      NULL      ;ASSIGN DELAY TO CRLF
384 0181 FE50      CPI      'P'
385 0183 0A13C3    JZ      PPRG      ;PROGRAM A PROM
386 0186 FE52      CPI      'R'
387 0188 0A40C6    JZ      R          ;EXAMINE AND CHANGE CPU REGISTERS
388 018B FE53      CPI      'S'
389 018D 0CA8C4    CZ      SPEED      ;SET PRINTER SPEED
390 0190 FE54      CPI      'T'
391 0192 0AC7C2    JZ      PROM      ;TRANSFER PROM TO RAM
392 0195 FE56      CPI      'V'
393 0197 0A56C3    JZ      TAPEL      ;WITH AREA OF RAM.
394 019A FE57      CPI      'W'
395 019C 0CD2C3    CZ      WTAPE      ;WRITE FILE TO MAG TAPE
396 019F 032EC1    JMP      MAIN
397 01A2          ;
398 01A2          ;NULL (N) SET DELAY FOR CARRIAGE RETURN
399 01A2          ;SENT TO CONSOLE DEVICE.
400 01A2          ;
401 01A2 0DB0C1    NULL:    CALL      ECHO      ;GET KB CHAR AND DISPLAY IT.
402 01A5 D630      SUI      '0'
403 01A7 FE0A      CPI      10
404 01A9 D223C4    JNC      ERROR
405 01AC 3200D3    STA      NULOC      ;STORE IN SCRATCHRAM
406 01AF C9        RET
407 01B0          ;
408 01B0          ;ECHO..GET A CHARACTER FROM THE CRT KEYBOARD
409 01B0          ;THEN SEND IT TO THE SCREEN. CHAR RETURNED IN A
410 01B0          ;
411 01B0 05        ECHO:    PUSH     B          ;SAVE B-C
412 01B1 0D87C3    CALL     INC          ;GET KB CHAR.
413 01B4 0D57C4    CALL     LOCON      ;CONVERT LC TO UC
414 01B7 4F        MOV      C,A          ;PUT IN REG. C
415 01B8 0DB1C2    CALL     OUTC      ;SEND IT TO SCREEN.
416 01BB 01        POP      B          ;RESTORE B-C
417 01BC C9        RET          ;RETURN.
418 01BD          ;
419 01BD          ;MESSAGE AND CONSTANT STORAGE AREA.
420 01BD          ;
421 01BD 53504143    MS001:    DB          'SPACE BYTE 8085 MONITOR'
      01C1 45204259
      01C5 54452038
      01C9 30383520
      01CD 4D4F4E49
      01D1 544F32

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

322 C1D4 20562D31          DB      ' V-1.0  TOP OF RAM AT '
      C1D8 2E302020
      C1DC 544F5020
      C1E0 4F462052
      C1E4 414D2041
      C1E8 5420
323 C1EA 002D      SZ001  EQU      $-MS001
324 C1EA 0A0D0000  CRLFDB: DB      10,13,0,0
325 C1EE 00000000          DB      0,0,0,0
326 C1F2 000000          DB      0,0,0
327 C1F5 43484543  MS002: DB      'CHECKSUM ERROR'
      C1F9 4B53554D
      C1FD 20455252
      C201 4F52
328 C203 000E      SZ002  EQU      $-MS002
329 C203 46494C45  MS003: DB      'FILE SIZE ERROR'
      C207 2053495A
      C20B 45204552
      C20F 524F52
330 C212 000F      SZ003  EQU      $-MS003
331 C212 4449534B  MS004: DB      'DISK FULL'
      C216 2046554C
      C21A 4C
332 C21B 0009      SZ004  EQU      $-MS004
333 C21B 4449534B  MS005: DB      'DISK NOT READY'
      C21F 204E4F54
      C223 20524541
      C227 4459
334 C229 000E      SZ005  EQU      $-MS005
335 C229 4B455942  MS006: DB      'KEYBOARD ERROR!'
      C22D 4F415244
      C231 20455252
      C235 4F522107
336 C239 0010      SZ006  EQU      $-MS006
337 C239 435254      MS007: DB      'CRT'
338 C23C 0003      SZ007  EQU      $-MS007
339 C23C 5052494E  MS008: DB      'PRINTER'
      C240 544552
340 C243 0007      SZ008  EQU      $-MS008
341 C243 204C4953  MS009: DB      ' LIST DEVICE = '
      C247 54204445
      C24B 56494345
      C24F 203D20
342 C252 000F      SZ009  EQU      $-MS009
343 C252 204D454D  MS010: DB      ' MEMORY WRITE ERRORS'
      C256 4F525920
      C25A 57524954
      C25E 45204552
      C262 524F5253
344 C266 0014      SZ010  EQU      $-MS010
345 C266 54415045  MS011: DB      'TAPE VERIFIED'
      C26A 20564552
      C26E 49464945
      C272 44
346 C273 000D      SZ011  EQU      $-MS011

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

347 C273 4E4F20 MS014: DB 'NO '
348 C276 4449534B MS013: DB 'DISK'
349 C27A 0004 SZ013 EQU $-MS013
350 C27A 0007 SZ014 EQU $-MS014
351 C27A 4E4F204D MS015: DB 'NO MAG TAPE'
    C27E 41472054
    C282 415045
352 C285 000B SZ015 EQU $-MS015
353 C285 444F4553 MS016: DB 'DOES NOT COMPARE'
    C289 204E4F54
    C28D 20434F4D
    C291 50415245
354 C295 0010 SZ016 EQU $-MS016
355 C295 444F4E45 MS017: DB 'DONE'
356 C299 0004 SZ017 EQU $-MS017
357 C299 ;
358 C299 ;CRLF SEND CRLF TO LISTING DEVICE
359 C299 ;
360 C299 05 CRLF: PUSH B ;SAVE REGISTERS
361 C29A 0607 MVI B,7
362 C29C 0E0D MVI C,13
363 C29E 0D93C5 CALL OUTP ;SEND IT TO PRINTER
364 C2A1 0E0A MVI C,10
365 C2A3 0D93C5 CALL OUTP
366 C2A6 0E00 MVI C,0 ;SEND FILL CHARACTERS
367 C2A8 0D93C5 CRLF: CALL OUTP
368 C2AB 05 DCR B
369 C2AC 02A8C2 JNZ CRLF
370 C2AF 01 POP B
371 C2B0 09 RET
372 C2B1 ;
373 C2B1 ;OUTC SENDS ONE CHARACTER TO CRT
374 C2B1 ;
375 C2B1 3A07D8 OUTC: LDA CSTAT ;SEE WHO IS CONSOLE
376 C2B4 A7 ANA A
377 C2B5 CABCC2 JZ OUTC1
378 C2B8 79 MOV A,C
379 C2B9 03F2C7 JMP VIDEO
380 C2BC DBC1 OUTC1: IN CSTAT ;GET STATUS OF USART
381 C2BE E601 ANI TXRDY ;CHECK TX BUFFER FULL?
382 C2C0 CAB1C2 JZ OUTC ;FULL, LOOP AND WAIT.
383 C2C3 79 MOV A,C ;GET CHAR FROM REG. C
384 C2C4 D3C0 OUT SDATA ;PUT INTO USART
385 C2C6 09 RET ;RETURN
386 C2C7 ;
387 C2C7 3206D8 PROM: STA ERBYT ;SAVE CONTROL CHAR
388 C2CA CD12C4 CALL STRING
389 C2CD D1 POP D ;HI LIMIT ADDRESS
390 C2CE E1 POP H ;LO STARTING ADDRESS
391 C2CF 010000 LXI B,0 ;PROM ADDRESS
392 C2D2 CDE3C2 PROM1: CALL RPRM
393 C2D5 CD08C4 CALL LIMIT ;ARE WE DONE?
394 C2D8 D2D2C2 JNC PROM1 ;NO
395 C2DB 2195C2 LXI H,MS017 ;YES, DISPLAY
396 C2DE 1E04 MVI E,SZ017 ;"DONE.."

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

397 0250 0327C4      JMP      CHK1
398 0253      ;
399 0253 3E24      RPRM:  MVI      A,READP
400 0255 D3DB      OUT      PORTC
401 0257 79      MOV      A,C      ;LO-ORDER FROM ADDRESS
402 0259 2F      CMA      ;COMPLEMENT
403 0259 D3DA      OUT      PORTB      ;APPLY TO PROM.
404 025B 78      MOV      A,B      ;HI-ORDER FROM ADDRESS
405 025C 2F      CMA      ;COMPLIMENT
406 025D D3D9      OUT      PORTA      ;APPLY TO PROM
407 025F 03      INX      B      ;INCREMENT FROM ADDRESS
408 02F0 DBCC      IN      PORTD      ;READ BYTE FROM PROM
409 02F2 323CD8     STA      TEMP      ;STORE TEMP
410 02F5 97      SUB      A
411 02F6 D3D9      OUT      PORTA
412 02F8 D3DB      OUT      PORTC
413 02FA 3A06D8     LDA      ERBYT      ;GET CONTROL CHAR
414 02FD FE43      CPI      'C'      ;COMPARE?
415 02FF CA0DC3     JZ      PROM2      ;YES
416 0302 FE54      CPI      'T'      ;TRANSFER?
417 0304 0222C4     JNZ      ERROR      ;MUST BE ONE OR THE OTHER
418 0307      ;
419 0307      ;TRANSFER FROM CONTENT TO RAM
420 0307      ;
421 0307 3A3CD8     LDA      TEMP
422 030A 2F      CMA
423 030B 77      MOV      M,A
424 030C 09      RET
425 030D      ;
426 030D      ;COMPARE FROM CONTENT TO RAM
427 030D      ;
428 030D 3A3CD8     PROM2:  LDA      TEMP
429 0310 2F      CMA
430 0311 BE      CMP      M
431 0312 0B      RZ
432 0313      ;
433 0313      ;ERROR IF NO MATCH
434 0313      ;
435 0313 2185C2     LXI      H,MS016 ;DISPLAY
436 0316 1E10      MVI      E,SZ016 ;"DOES NOT COMPARE"
437 0318 0327C4     JMP      CHK1
438 031B      ;
439 031B 0D12C4     PPRG:  CALL     STRING
440 031E D1      POP      D
441 031F E1      POP      H
442 0320 223CD8     SHLD     TEMP
443 0323 3E64      MVI      A,100
444 0325 3206D8     STA      ERBYT
445 0328 3E0E      MVI      A,WRTP
446 032A D3DB      OUT      PORTC
447 032C 3A3CD8     PRG1:  LHLD     TEMP
448 032F 010000     LXI      B,0
449 0332 0D52C3     PRG2:  CALL     PRGA
450 0335 0B03C4     CALL     LIMIT
451 0338 D232C3     JNC      PRG2

```

SPACE BYTE 8085 PROM MONITOR 11-14-77

```

452 033B 3A06D8      LDA      ERBYT
453 033E 3D          DCR      A
454 033F 3206D8      STA      ERBYT
455 0342 0220C3      JNZ      PROG1
456 0345 97          SUB      A
457 0346 D3D9        OUT      PORTA
458 0348 D3DB        OUT      PORTC
459 034A 2195C2      LXI      H,MS017
460 034D 1E04        MVI      E,SZ017
461 034F 0327C4      JMP      CHK1
462 0352              ;
463 0352 7E          PROGA: MOV      A,M
464 0353 2F          CMA
465 0354 D3DA        OUT      PORTB
466 0356 DBDB        IN       PORTC
467 0358 F601        ORI      STB
468 035A D3DB        OUT      PORTC
469 035C E6FE        ANI      255-STB
470 035E D3DB        OUT      PORTC
471 0360 79          MOV      A,C
472 0361 2F          CMA
473 0362 D3DA        OUT      PORTB
474 0364 78          MOV      A,B
475 0365 2F          CMA
476 0366 D3D9        OUT      PORTA
477 0368 03          INX      B
478 0369 DBDB        IN       PORTC
479 036B F610        ORI      PPULS
480 036D D3DB        OUT      PORTC
481 036F CD76C3      CALL     DLAY
482 0372 E6EF        ANI      255-PPULS
483 0374 D3DB        OUT      PORTC
484 0376              ;
485 0376 F5          DLAY:  PUSH     PSW
486 0377 C5          PUSH     B
487 0378 D5          PUSH     D
488 0379 E5          PUSH     H
489 037A 3E50        MVI      A,80
490 037C E1          DLOOP: POP      H
491 037D E5          PUSH     H
492 037E 3D          DCR      A
493 037F 0270C3      JNZ      DLOOP
494 0382 E1          POP      H
495 0383 D1          POP      D
496 0384 C1          POP      B
497 0385 F1          POP      PSW
498 0386 C9          RET
499 0387              ;
500 0387              ;INC  RECEIVES ONE CHARACTER FROM CRT
501 0387              ;CHARACTER RETURNED IN A  W/FLAGS SET.
502 0387              ;
503 0387 3A07D8      INC:   LDA      CSTAT  ;SEE WHO IS CONSOLE
504 038A A7          ANA      A
505 038B 0A9AC3      JZ      INC1
506 038E DBF9        INC2:  IN       0F9H  ;CHECK KEYBOARD STATUS

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

507 0390 E601      ANI      1
508 0392 C28EC3    JNZ      INC2      ;LOOP AND WAIT
509 0395 DBF8      IN       0F8H      ;GET CHAR.
510 0397 E67F      ANI      7FH      ;MASK TO 7 BITS
511 0399 C9        RET
512 039A DBC1      INC1:  IN       SSTAT  ;GET STATUS OF USART
513 039C E602      ANI      RMRDY    ;CHECK RX CHAR. PRESENT?
514 039E C987C3    JZ       INC      ;NOT HERE YET, WAIT.
515 03A1 DBC0      IN       SDATA    ;GET CHAR FROM USART
516 03A3 E67F      ANI      127     ;MASK TO 7 BITS/SET FLAGS
517 03A5 C9        RET              ;RETURN.
518 03A6          ;
519 03A6          ;MSOUT MESSAGE OUTPUT ROUTINE
520 03A6          ;START OF STRING IS IN H-L
521 03A6          ;STRING LENGTH IS IN REG. E
522 03A6          ;
523 03A6 4E        MSOUT: MOV     C,M      ;GET STRING CHAR.
524 03A7 CDB1C2    CALL    DUTC      ;SEND TO CRT.
525 03AA 23        INX     H          ;INCREMENT MSG POINTER
526 03AB 1D        DCR     E          ;DECREMENT CHAR. COUNT
527 03AC C2A6C3    JNZ     MSOUT     ;LOOP, IF NOT DONE.
528 03AF C9        RET              ;RETURN.
529 03B0          ;
530 03B0          ;CALCULATE TOP OF RAM MEMORY
531 03B0          ;START LOCATION IS IN H-L
532 03B0          ;
533 03B0 7E        HILOC: MOV     A,M      ;GET DATA FROM RAM
534 03B1 2F        CMA          ;COMPLEMENT DATA
535 03B2 77        MOV     M,A        ;WRITE IT BACK TO SAME LOC.
536 03B3 3E        CMP     M          ;MAKE SURE IT GOT THERE.
537 03B4 C2BDC3    JNZ     EXIT1     ;NOT READ/WRITE MEMORY
538 03B7 2F        CMA          ;COMPLEMENT AND
539 03B8 77        MOV     M,A        ;RESTORE ORIGINAL CONTENTS.
540 03B9 24        INR     H          ;INCREMENT H (HI-ORDER)
541 03BA C3B0C3    JMP     HILOC     ;KEEP GOING.
542 03BD          ;
543 03BD 7C        EXIT1: MOV     A,H      ;TEST FOR NO
544 03BE A7        ANA     A          ;RAM AT ALL.
545 03BF CAC3C3    JZ       EXIT2
546 03C2 2B        DCR     H          ;SUBTRACT 1 FROM 16 BITS
547 03C3 2201D8    EXIT2: SHLD    HIBYTE ;STORE IN SCRATCHRAM
548 03C6 E5        PUSH   H
549 03C7 CD4AC5    CALL    CRLF      ;CRLF TO CRT
550 03CA 21BDC1    LXI     H,MS001  ;MONITOR GREETING MESSAGE.
551 03CD 1E2D      MVI     E, SZ001  ;SIZE TO REG. E
552 03CF CDA6C3    CALL    MSOUT     ;DISPLAY MESSAGE
553 03D2 E1        POP     H          ;RESTORE H-L
554 03D3 7C        MOV     A,H      ;GET USER STACK
555 03D4 3226D8    STA     SLLOC
556 03D7 7D        MOV     A,L
557 03D8 3227D8    STA     SLLOC+1
558 03DB CD6FC4    CALL    DISAD     ;DISPLAY HI MEMORY VALUE
559 03DE C34AC5    JMP     CRLF      ;RETURN THRU CRLF.
560 03E1          ;
561 03E1          ;MEMTP READ TOP OF RAM MEMORY ADDRESS

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

562 03E1      ;
563 03E1 3A02D8 MEMTP: LDA      HIBYTE+1
564 03E4 47      MOV      B,A      ;PUT HI ADDR BYTE IN REG. B
565 03E5 3A01D8      LDA      HIBYTE ;LO ADDR BYTE IN REG. A
566 03E8 09      RET
567 03E9      ;
568 03E9      ;HEXAS  HEXIDECIMAL ARITHMETIC
569 03E9      ;TYPE TWO NUMBERS AND THE SUM AND
570 03E9      ;DIFFERENCE ARE DISPLAYED ON THE
571 03E9      ;CONSOLE DEVICE.
572 03E9      ;
573 03E9 0D12C4  HEXAS: CALL    STRING ;GET TWO NUMBERS
574 03EC 0D4AC5      CALL    CRLF
575 03EF D1      POP      D      ;NUMBER TWO
576 03F0 E1      POP      H      ;NUMBER ONE
577 03F1 E5      PUSH     H      ;SAVE NUMBER ONE
578 03F2 19      DAD      D      ;ADD THE TWO
579 03F3 0D6FC4      CALL    DISAD ;DISPLAY THE SUM
580 03F6 0E20      MVI      C,' ' ;SEND A SPACE
581 03F8 0DB1C2      CALL    OUTC  ;TO CRT
582 03FB E1      POP      H      ;GET NUMBER TWO
583 03FC 7D      MOV      A,L      ;SUBTRACT #1
584 03FD 93      SUB      E      ;FROM #2
585 03FE 6F      MOV      L,A
586 03FF 7C      MOV      A,H
587 0400 9A      SBB      D
588 0401 67      MOV      H,A
589 0402 0D6FC4      CALL    DISAD ;DISPLAY DIFFERENCE
590 0405 034AC5      JMP      CRLF
591 0408      ;
592 0408      ;LIMIT  TEST TO SEE IF D-E EQUAL TO H-L
593 0408      ;
594 0408 23      LIMIT: INX      H      ;INCREMENT H-L
595 0409 7C      MOV      A,H      ;GET HI BYTE
596 040A B5      ORA      L      ;OR WITH LO BYTE
597 040B 37      STC      ;SET CARRY FOR SUBTRACT
598 040C 08      RZ      ;H-L = 0
599 040D 7B      MOV      A,E      ;SUBTRACT
600 040E 95      SUB      L      ;H-L FROM D-E
601 040F 7A      MOV      A,D      ;AND LOOK FOR CARRY
602 0410 9C      SBB      H      ;TO BE RESET IF
603 0411 09      RET      ;H-L > D-E
604 0412      ;
605 0412      ;STRING  GET A COMMAND FROM THE KEYBOARD
606 0412      ;AND STORE THE 16 BIT NUMBERS ON THE STACK.
607 0412      ;
608 0412 0D33C4  STRING: CALL    GET16 ;GET 16 BIT NUMBER IN H-L
609 0415 E3      XTHL      ;EXCHANGE WITH TOP OF
610 0416 E5      PUSH     H      ;STACK AND REPLACE RETURN ADDR.
611 0417 0D      DCR      C      ;DECREMENT GROUP COUNT
612 0418 D21FC4      JNC      STRN0
613 0419 0222C4      JNZ      ERROR ;NOT ENOUGH NUMBERS!!
614 041E 09      RET      ;ALL CORRECT
615 041F 0212C4  STRN0: JNZ      STRING ;GET ANOTHER GROUP
616 0422      ;

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

617 C422      ;DISPLAY KEYIN ERROR ROUTINE
618 C422      ;
619 C422 2129C2 ERROR: LXI      H,MS006 ;DISPLAY "KEYBOARD ERROR"
620 C425 1E10      MVI      E,SZ006
621 C427 CD4AC5 CHK1:  CALL    CRLF
622 C429 CDA6C3      CALL    MSOUT
623 C42D 31FFD8      LXI      SP,STACK
624 C430 C32EC1      JMP      MAIN
625 C433      ;
626 C433      ;GET16  ONE TO FOUR ASCII CHARACTERS
627 C433      ;RECEIVED FROM THE KEYBOARD ARE CONVERTED
628 C433      ;TO 16 BIT NUMBERS IN H-L
629 C433      ;
630 C433 CD60C4 GET16: CALL    DELIM  ;GET CHAR AND TEST FOR
631 C436 CA22C4      JZ      ERROR  ;FOR TERMINATOR.
632 C439 210000 GET0:  LXI      H,0      ;PRESET H-L = 0
633 C43C 47      GET1:  MOV      B,A
634 C43D CDAFC9      CALL    NBL     ;CONVERT TO 4-BITS
635 C440 DA4FC4      JC      GET2    ;NOT NUMBER, TEST FOR END.
636 C443 29      DAD      H         ;SHIFT NUMBER IN
637 C444 29      DAD      H         ;H-L 4 POSITIONS TO
638 C445 29      DAD      H         ;THE LEFT. MAKE ROOM
639 C446 29      DAD      H         ;FOR THIS NUMBER.
640 C447 B5      ORA      L         ;PUT THESE 4 BITS
641 C448 6F      MOV      L,A       ;IN AS LSB.
642 C449 CDB0C1      CALL    ECHO    ;GET NEXT ASCII CHAR.
643 C44C C33CC4      JMP      GET1    ;LOOP
644 C44F 73      GET2:  MOV      A,B  ;GET BACK CHAR.
645 C450 CD63C4      CALL    DEL1    ;TEST FOR TERMINATOR
646 C453 C222C4      JNZ      ERROR
647 C456 C9      RET
648 C457      ;
649 C457      ;LOCON  LOWER CASE TO UPPER CASE CONVERTER
650 C457      ;
651 C457 FE61      LOCON: CPI      61H
652 C459 F8      RM
653 C45A FE7B      CPI      7BH
654 C45C F0      RP
655 C45D E6DF      ANI      0DFH
656 C45F C9      RET
657 C460      ;
658 C460      ;DELIM  GET CHAR$ ECHO IT, AND TEST FOR DELIMITER
659 C460      ;
660 C460 CDB0C1      DELIM: CALL    ECHO    ;GET CHAR/DISPLAY IT
661 C463 FE20      DEL1:  CPI      /
662 C465 C9      RZ
663 C466 FE2C      CPI      /
664 C468 C9      RZ
665 C469 FE0D      CPI      13
666 C46B 37      STC
667 C46C C9      RZ
668 C46D 3F      CMC
669 C46E C9      RET
670 C46F      ;
671 C46F      ;DISAD  DISPLAY MEMORY ADDRESS.  THE 16-BIT

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

672 046F      ;NUMBER IN H-L IS DISPLAYED AS 4 ASCII CHARS.
673 046F      ;
674 046F 7C    DISAB: MOV     A,H      ;DISPLAY HI BYTE
675 0470 CD77C4 CALL     DIS8
676 0473 7D    MOV     A,L      ;DISPLAY LO BYTE
677 0474 C377C4 JMP      DIS8
678 0477      ;
679 0477      ;DIS8  DISPLAY 8 BIT VALUE IN A AS
680 0477      ;TWO ASCII CHARACTERS.
681 0477      ;
682 0477 F5    DIS8:  PUSH     PSW
683 0478 0F    RRC              ;GET HI NIBBLE
684 0479 0F    RRC
685 047A 0F    RRC
686 047B 0F    RRC
687 047C CD99C4 CALL     HTOA      ;CONVERT HEX TO ASCII
688 047F CDB1C2 CALL     OUTC      ;SEND TO PRINTER
689 0482 F1    POP     PSW      ;GET BYTE FOR LO NIBBLE
690 0483 CD99C4 CALL     HTOA
691 0486 C3B1C2 JMP      OUTC
692 0489      ;
693 0489      ;LIST  LIST OF VALID ASCII CHARACTERS
694 0489      ;USED IN HEXIDECIMAL NUMBER NOTATION.
695 0489      ;
696 0489 30313233 LIST:  DB      '0123456789ABCDEF'
        048D 34353637
        0491 38394142
        0495 43444546
697 0499      ;
698 0499      ;HTOA  HEX TO ASCII CONVERTER
699 0499      ;ASCII CHAR IS RETURNED IN REG. C
700 0499      ;
701 0499 E5    HTOA:  PUSH     H
702 049A D5    PUSH     D
703 049B 2189C4 LXI      H,LIST
704 049E E60F ANI      15
705 04A0 5F    MOV     E,A
706 04A1 1600 MVI      D,0
707 04A3 19    DAD     D
708 04A4 4E    MOV     C,M
709 04A5 D1    POP     D
710 04A6 E1    POP     H
711 04A7 C9    RET
712 04A8      ;
713 04A8      ;SPEED SET BAUD RATE FOR THE PRINTER
714 04A8      ;
715 04A8 CD33C4 SPEED:  CALL     GET16   ;GET A NUMBER
716 04AB CD4AC5 CALL     ORLF   ;ORLF TO CRT
717 04AE 7C    MOV     A,H      ;GET HI ORDER 2 NUMBERS
718 04AF 1F    RAR
719 04B0 1F    RAR              ;SHIFT FOR TEST
720 04B1 1F    RAR
721 04B2 1F    RAR
722 04B3 E60F ANI      15      ;MASK FOR 4 BITS
723 04B5 CAF7C4 JC      322      ;FIRST DIGIT IS A 0.

```


SPACE BYTE 8085 PROM MONITOR 11-14-77

```

724 C4B8 FE09          CPI      9          ;NO, IS IT 9
725 C4BA C203C4        JNZ      SP3        ;NO!
726 C4BD 211440        LXI      H,SP600    ;YES, SET SPEED
727 C4C0 C310C5        JMP      SP4        ;TO 9600 BPS.
728 C4C3 FE04          SP3: CPI      4          ;IS IT 4
729 C4C5 C203C4        JNZ      SP5        ;NO!
730 C4C8 212840        LXI      H,S4800    ;YES, SET SPEED
731 C4CB C310C5        JMP      SP4        ;TO 4800 BPS.
732 C4CE FE02          SP5: CPI      2          ;IS IT A 2?
733 C4D0 C2D9C4        JNZ      SP6        ;NO!
734 C4D3 215040        LXI      H,S2400    ;YES, SET SPEED
735 C4D6 C310C5        JMP      SP4        ;TO 4800 BPS
736 C4D9 FE01          SP6: CPI      1          ;IS IT 1?
737 C4DB C222C4        JNZ      ERROR      ;KEY ENTRY ERROR.
738 C4DE 7C            MOV      A,H        ;YES IS A 1
739 C4DF E60F          ANI      15         ;GET NEXT DIGIT
740 C4E1 FE08          CPI      8          ;IS SECOND NUMBER 8?
741 C4E3 C2E0C4        JNZ      SP11       ;NO, THEN SHOULD BE 2
742 C4E6 216B40        LXI      H,S1800    ;YES, SET SPEED
743 C4E9 C310C5        JMP      SP4        ;TO 1800 BPS.
744 C4EC FE02          SP11: CPI      2          ;IS IT 2?
745 C4EE C222C4        JNZ      ERROR      ;KEY ENTRY ERROR
746 C4F1 21A040        LXI      H,S1200    ;YES, SET SPEED
747 C4F4 C310C5        JMP      SP4        ;TO 1200 BPS
748 C4F7 7C            SP2: MOV      A,H        ;GET 1ST 2 NUMBERS
749 C4F8 FE03          CPI      3          ;IS 2ND = A 3?
750 C4FA C203C5        JNZ      SP9        ;NO!
751 C4FD 219042        LXI      H,S0300    ;YES, SET SPEED
752 C500 C310C5        JMP      SP4        ;TO 300 BPS
753 C503 FE01          SP9: CPI      1          ;IS 2ND = A 1?
754 C505 C222C4        JNZ      ERROR      ;KEY ENTRY ERROR
755 C508 7D            SP7: MOV      A,L        ;GET LO-ORDER NUMBERS
756 C509 FE50          CPI      50H        ;IS 3RD NUMBER 5
757 C50B C214C5        JNZ      SP10       ;NO!
758 C50E 210045        LXI      H,S0150    ;YES, SET SPEED
759 C511 C310C5        JMP      SP4        ;TO 150 BPS
760 C514 FE10          SP10: CPI      10H        ;IS 3RD NUMBER 1?
761 C516 C222C4        JNZ      ERROR      ;NO, KEY ENTRY ERROR
762 C519 21D146        LXI      H,S0110    ;YES, SET SPEED
763 C51C 2204D8        SP4: SHLD     BAUDP    ;TO 110 BAUD
764 C51F C9            RET
765 C520              ;
766 C520              ;PRNAD PRINT MEMORY ADDRESS. THE
767 C520              ;16 BIT NUMBER IN H-L IS PRINTED AS 4
768 C520              ;ASCII CHARACTERS.
769 C520              ;
770 C520 7C            PRNAD: MOV      A,H
771 C521 CD81C5        CALL     BYTEP
772 C524 7D            MOV      A,L
773 C525 CD81C5        JMP      BYTEP
774 C528              ;
775 C528              ;FILL FILL MEMORY WITH A CONSTANT
776 C528              ;USER SPECIFIES LO-ADDR/HI-ADDR/CONSTANT.
777 C528              ;
778 C528 0C            FILL: INR      C          ;LOOK FOR 3 NUMBERS

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

779 C529 CD1204      CALL    STRING    ;GET 3 NUMBERS
780 C52C CD4A05      CALL    CRLF
781 C52F C1          POP     B          ;CONSTANT IS REG C.
782 C530 D1          POP     D          ;HI MEMORY ADDR.
783 C531 E1          POP     H          ;LO MEMORY ADDR.
784 C532 71          FILL1: MOV     M,C    ;WRITE TO MEMORY
785 C533 CD08C4      CALL    LIMIT    ;SEE IF WERE DONE.
786 C536 D232C5      JNC     FILL1    ;NO, CONTINUE
787 C539 C9          RET
788 C53A             ;
789 C53A             ;MOVEB  MOVE A BLOCK OF MEMORY
790 C53A             ;
791 C53A 0C          MOVEB: INR     C      ;GET THREE NUMBERS
792 C53B CD1204      CALL    STRING
793 C53E C1          POP     B          ;NEW ORG
794 C53F D1          POP     D          ;OLD TOP ADDR.
795 C540 E1          POP     H          ;OLD ORG
796 C541 7E          MOVE1: MOV     A,M    ;GET BYTE FROM OLD
797 C542 02          STAX    B          ;WRITE TO NEW
798 C543 03          INX     B          ;INCREMENT NEW
799 C544 CD08C4      CALL    LIMIT    ;INCREMENT OLD
800 C547 D241C5      JNC     MOVE1    ;AND LOOP IF NOT DONE.
801 C54A             ;
802 C54A             ;CRLF SENDS CARRIAGE RETURN LINE FEED TO CONSOLE DEVICE.
803 C54A             ;
804 C54A E5          CRLF: PUSH    H      ;SAVE CURRENT
805 C54B D5          PUSH    D          ;CPU REGISTER
806 C54C F5          PUSH    PSW       ;CONTENTS.
807 C54D 21EAC1      LXI     H,CRLF    ;POINT TO CRLF DATA BLOCK
808 C550 3A00D8      LDA     MULLC    ;GET NUMBER OF NULLS TO ADD
809 C553 C602        ADI     2         ;ADD 2 FOR CRLF
810 C555 5F          MOV     E,A       ;PUT LENGTH IN REG. E
811 C556 CDA6C3      CALL    MSGOUT    ;SEND THE MESSAGE
812 C559 F1          POP     PSW       ;RESTORE
813 C55A D1          POP     D          ;CPU REGISTER CONTENTS
814 C55B E1          POP     H          ;AND
815 C55C C9          RET              ;RETURN TO CALLER
816 C55D             ;
817 C55D             ;DUMP  THIS ROUTINE DISPLAYS MEMORY
818 C55D             ;CONTENTS ON THE LISTING DEVICE.
819 C55D             ;
820 C55D CD1204      DUMP:  CALL    STRING ;GET 2 NUMBERS
821 C560 D1          POP     D          ;HI ADDR.
822 C561 E1          POP     H          ;LO ADDR.
823 C562 CD99C2      DUMP1: CALL    CRLF  ;CRLF TO LISTING DEVICE
824 C565 CD20C5      CALL    PRNAD    ;DISPLAY ADDR.
825 C568 0E20        DUMP2: MVI     C,' ' ;SEND SPACE
826 C56A CD93C5      CALL    OUTP    ;TO LIST DEVICE
827 C56D 7E          MOV     A,M       ;GET MEMORY BYTE
828 C56E CD81C5      CALL    BYTDP    ;SEND TO PRINTER
829 C571 CD08C4      CALL    LIMIT    ;SEE IF WERE DONE
830 C574 DA99C2      JC      CRLF     ;DONE
831 C577 00          NOP              ;DO SOME MORE.
832 C578 7D          MOV     A,L
833 C579 E60F        ANI     15

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

834 C57B C269C5      JNZ      DUMP2
835 C57E C362C5      JMP      DUMP1
836 C581             ;
837 C581             ;BYTER PRINT A BYTE OF MEMORY
838 C581             ;ON THE LISTING DEVICE AS TWO
839 C581             ;ASCII CHARACTERS
840 C581             ;
841 C581 F5          BYTER:  PUSH    PSW      ;SAVE A COPY
842 C582 0F          RRC      ;SHIFT
843 C583 0F          RRC      ;FOR
844 C584 0F          RRC      ;HI
845 C585 0F          RRC      ;ORDER
846 C586 CD99C4      CALL     HTOA      ;CONVERT TO ASCII
847 C589 CD93C5      CALL     OUTP      ;SEND TO PRINTER
848 C58C F1          POP      PSW      ;GET ORIGINAL
849 C58D CD99C4      CALL     HTOA      ;CONVERT TO ASCII
850 C590 C393C5      JMP      OUTP      ;SEND TO PRINTER
851 C593             ;
852 C593             ;OUTP SENDS ONE CHAR TO LISTING DEVICE
853 C593             ;CHARACTER IS EXPECTED IN REG. C
854 C593             ;
855 C593 3A03D8      OUTP:  LDA      PSTAT   ;CHECK PRINTER
856 C596 A7          ANA      A          ;STATUS
857 C597 CAB1C2      JZ       OUTC      ;SEND DATA TO CRT
858 C59A             ;
859 C59A             ;WAIT FOR USART TXR TO EMPTY
860 C59A             ;
861 C59A C5          PUSH     B
862 C59B E5          PUSH     H
863 C59C 060A      OUTP2:  MVI      B,10
864 C59E DB01      OUTP1:  IN       SSTAT   ;GET USART STATUS
865 C5A0 E604      ANI      4          ;TX EMPTY?
866 C5A2 CA90C5      JZ       OUTP2      ;NO
867 C5A5 05          DCR      B          ;YES, MAYBE
868 C5A6 C29EC5      JNZ      OUTP1      ;CHECK 9 MORE TIMES
869 C5A9             ;
870 C5A9             ;SWITCH THE OUTPUT CONNECTIONS TO PRINTER
871 C5A9             ;
872 C5A9 3E27      MVI      A,27H
873 C5AB D3C1      OUT      SSTAT
874 C5AD             ;
875 C5AD             ;SWITCH THE SPEED TO MATCH THAT
876 C5AD             ;OF THE PRINTER IN USE
877 C5AD             ;
878 C5AD 2A04D8      LHLD     BAUDP
879 C5B0 7D          MOV      A,L
880 C5B1 D3DC      OUT      TIMLO
881 C5B3 7C          MOV      A,H
882 C5B4 D3DD      OUT      TIMHI
883 C5B6 3E0F      MVI      A,00FH
884 C5B8 D3D8      OUT      PORTI
885 C5BA             ;
886 C5BA             ;SEND CHARACTER
887 C5BA             ;
888 C5BA 79          MOV      A,C

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

889 C5BB D3C0          OUT      SDATA
890 C5BD              ;
891 C5BD              ;RESTORE EVERYTHING THE WAY IT WAS
892 C5BD              ;
893 C5BD 060A          OUTP3: MVI      B,10
894 C5BF DBC1          OUTP4: IN       SSTAT
895 C5C1 E604          ANI       4
896 C5C3 CABDC5        JZ        OUTP3
897 C5C6 05           DCR       B
898 C5C7 C2BFC5        JNZ       OUTP4
899 C5CA              ;
900 C5CA 3E25          MVI       A,25H
901 C5CC D3C1          OUT      SSTAT
902 C5CE              ;
903 C5CE 2A19D8        LHLD      BAUDC
904 C5D1 7D           MOV       A,L
905 C5D2 D3DC          OUT      TIMLO
906 C5D4 7C           MOV       A,H
907 C5D5 D3DD          OUT      TIMHI
908 C5D7 3ECF          MVI       A,0CFH
909 C5D9 D3D8          OUT      PORTI
910 C5DB E1           POP       H
911 C5DC C1           POP       B
912 C5DD              ;
913 C5DD C9           RET
914 C5DE              ;
915 C5DE              ;INSPECT  INSPECT THE CONTENTS OF MEMORY
916 C5DE              ;ONE LOCATION AT A TIME.  CHANGE CONTENTS
917 C5DE              ;IF NECESSARY.
918 C5DE              ;
919 C5DE CD33C4        INSPECT:  CALL    GET16  ;GET AN ADDR.
920 C5E1 DA2EC1        JC        MAIN    ;OR ENTERED
921 C5E4 7E           INSP1:  MOV       A,M    ;GET CONTENTS OF MEMLOC
922 C5E5 CD77C4        CALL    DIS8     ;DISPLAY CONTENTS
923 C5E8 0E2D          MVI       C,'-'    ;DISPLAY A DASH
924 C5EA CDB1C2        CALL    OUTC     ;
925 C5ED CD60C4        CALL    DELIM    ;GET A CHAR
926 C5F0 DA2EC1        JC        MAIN    ;OR ENTERED
927 C5F3 CAFFC5        JZ        INSP2   ;SPACE ENTERED DISPLAY NEXT
928 C5F6 EB           XCHG          ;GET TWO ASCII CHARACTERS
929 C5F7 CD39C4        CALL    GET0     ;CONVERT TO BINARY AND
930 C5FA EB           XCHG          ;
931 C5FB 73           MOV       M,E      ;STORE IN MEMORY
932 C5FC DA2EC1        JC        MAIN    ;RETURN ON CR
933 C5FF 23           INSP2:  INX       H      ;INCREMENT MEMORY POINTER
934 C600 C3E4C5        JMP       INSP1   ;DO ANOTHER
935 C603              ;
936 C603              ;ASSIGN LISTING DEVICE
937 C603              ;
938 C603 2143C2        ASSIGN:  LXI     H,MS009 ;DISPLAY
939 C606 1E0F          MVI       E,SZ009 ;"LIST DEVICE = "
940 C608 CDA6C3        CALL    MSOUT
941 C60B CD87C3        CALL    INC      ;GET A CHAR AND DISP IT.
942 C60E FE43          CPI       0C      ;LIST = CRT?
943 C610 CA25C6        JZ        A31     ;YES!

```


SPACE BYTE 3035 FROM MONITOR 11-14-77

```

944 0613 FE0D      CPI      13      ;RETURN?
945 0615 CA29C6    JZ       ADISP   ;DISPLAY STATUS
946 0618 FE50      CPI      'P'    ;LIST = PRINTER?
947 061A C222C4    JNZ      ERROR   ;MUST BE CRT OR PRINTER
948 061D 3EFF      MVI      A,-1    ;SET STATUS = PRINTER
949 061F 3203D8    STA      PSTAT
950 0622 C329C6    JMP      ADISP
951 0625 97       AS1:    SUB      A      ;SET STATUS = CRT
952 0626 3203D8    STA      PSTAT
953 0629 3A03D8    ADISP:   LDA      PSTAT
954 062C A7        ANA      A
955 062D CA38C6    JZ       ADIS1
956 0630 213CC2    LXI      H,MS008
957 0633 1E07      MVI      E,SZ008
958 0635 C32AC4    JMP      CHK1+3
959 0638 2139C2    ADIS1:  LXI      H,MS007
960 063B 1E03      MVI      E,SZ007
961 063D C32AC4    JMP      CHK1+3
962 0640          ;
963 0640          ;R   EXAMINE AND MODIFY CPU REGISTERS
964 0640          ;
965 0640 213CC6    R:      LXI      H,RTBL ;POINT TO MATCH TABLE
966 0643 CD60C4    CALL     DELIM   ;GET A CHAR.
967 0646 DA8EC6    JC       R5      ;CR ENTERED
968 0649 0E0C      MVI      C,SZ0   ;TABLE SIZE
969 064B BE        R0:     CMP      M      ;COMPARE ENTRY TO TABLE
970 064C CA5AC6    JZ       R1      ;MATCH FOUND
971 064F 23        INX      H          ;NO MATCH, SKIP
972 0650 23        INX      H          ;TO NEXT TABLE
973 0651 23        INX      H          ;ENTRY.
974 0652 23        INX      H
975 0653 0D        DCR      C
976 0654 C248C6    JNZ      R0
977 0657 C322C4    JMP      ERROR
978 065A 0E20      R1:     MVI      C,' ' ;SEND A SPACE
979 065C CDB1C2    CALL     OUTC    ;TO THE CRT.
980 065F CDAAC6    R2:     CALL     DREG  ;DISPLAY REG. CONTENTS
981 0662 0E2D      MVI      C,'-'   ;SEND A DASH
982 0664 CDB1C2    CALL     OUTC    ;TO THE CRT.
983 0667 CD60C4    CALL     DELIM   ;GET A CHAR
984 066A DA2EC1    JC       MAIN    ;CR ENTERED
985 066D CA80C6    JZ       R4      ;SPACE ENTERED
986 0670 E5        PUSH     H
987 0671 C5        PUSH     B
988 0672 CD39C4    CALL     GET0    ;GET NEW VALUE
989 0675 7D        MOV      A,L
990 0676 12        STAX     D
991 0677 F1        POP      PSW
992 0678 B7        ORA      A
993 0679 FA7FC6    JM       R3
994 067C 1B        DCR      D
995 067D 7C        MOV      A,H
996 067E 12        STAX     D
997 067F E1        R3:     POP      H
998 0680 AF        R4:     XRA      A

```

SPACE BYTE 8085 PROM MONITOR 11-14-77

```

999 C681 B6          ORA      M
1000 C682 FA2EC1     JM       MAIN
1001 C685 78         MOV      A,B
1002 C686 FE0D       CPI      13
1003 C688 CA2EC1     JZ       MAIN
1004 C68B C35FC6     JMP      R2
1005 C68E CD4AC5     R5:     CALL  CRLF
1006 C691 0E20       R6:     MVI   C,' '
1007 C693 CDB1C2     CALL     DUTC
1008 C696 AF        XRA      A
1009 C697 B6        ORA      M
1010 C698 FA2EC1     JM       MAIN
1011 C69B 4E        MOV      C,M
1012 C69C CDB1C2     CALL     DUTC
1013 C69F 0E3D       MVI     C,'='
1014 C6A1 CDB1C2     CALL     DUTC
1015 C6A4 CDAC06     CALL     DREG
1016 C6A7 C391C6     JMP      R6
1017 C6AA           ;
1018 C6AA           ;DREG  DISPLAY REGISTER CONTENTS
1019 C6AA           ;
1020 C6AA 23        DREG:    INX      H
1021 C6AB 5E        MOV      E,M
1022 C6AC 23        INX      H
1023 C6AD 56        MOV      D,M
1024 C6AE 23        INX      H
1025 C6AF 46        MOV      B,M      ;SIZE 1 OR 2
1026 C6B0 23        INX      H      ;POINT TO NEXT
1027 C6B1 1A        LDAX     D
1028 C6B2 CD77C4     CALL     DISB    ;DISPLAY BYTE
1029 C6B5 05        DCR      B
1030 C6B6 F3        RM
1031 C6B7 13        INX      D
1032 C6B8 1A        LDAX     D
1033 C6B9 C377C4     JMP      DISB
1034 C6BC           ;
1035 C6BC           ;RTBL  REGISTER IDENTIFIER TABLE
1036 C6BC           ;
1037 C6BC 41        RTBL:    DB       'A'
1038 C6BD 1BD3       DW       ALDC
1039 C6BF 00        DB       0
1040 C6C0 42        DB       'B'
1041 C6C1 1CDB       DW       BLDC
1042 C6C3 00        DB       0
1043 C6C4 43        DB       'C'
1044 C6C5 1DD3       DW       CLDC
1045 C6C7 00        DB       0
1046 C6C8 44        DB       'D'
1047 C6C9 1ED3       DW       DLDC
1048 C6CB 00        DB       0
1049 C6CC 45        DB       'E'
1050 C6CD 1FD3       DW       ELDC
1051 C6CF 00        DB       0
1052 C6D0 46        DB       'F'
1053 C6D1 20DB       DW       FLDC

```


SPACE BYTE 8085 PROM MONITOR 11-14-77

```

1054 C6D3 00          DB      0
1055 C6D4 49          DB      'I'
1056 C6D5 21D8        DW      ILDC
1057 C6D7 00          DB      0
1058 C6D8 48          DB      'H'
1059 C6D9 22D8        DW      HLDC
1060 C6DB 00          DB      0
1061 C6DC 4C          DB      'L'
1062 C6DD 23D8        DW      LLDC
1063 C6DF 00          DB      0
1064 C6E0 4D          DB      'M'
1065 C6E1 22D8        DW      HLDC
1066 C6E3 01          DB      1
1067 C6E4 50          DB      'P'
1068 C6E5 24D8        DW      PLDC
1069 C6E7 01          DB      1
1070 C6E8 53          DB      'S'
1071 C6E9 26D8        DW      SLDC
1072 C6EB 01          DB      1
1073 C6EC FF          DB      -1
1074 C6ED 000C        SZR      EQU      (S-RTBL)/4
1075 C6ED              ;
1076 C6ED              ;
1077 C6ED              ;GO  EXECUTE USER PROGRAM
1078 C6ED              ;
1079 C6ED CD60C4        GO:      CALL      DELIM
1080 C6F0 CA37C7        JZ        GO2
1081 C6F3 CD39C4        CALL      GET0
1082 C6F6 2234D8        SHLD      EXIT
1083 C6F9 CA37C7        JZ        GO2
1084 C6FC 2126D8        GO1:      LXI      H,SLDC
1085 C6FF 56            MOV      D,M
1086 C700 23            INX      H
1087 C701 5E            MOV      E,M
1088 C702 D5            PUSH     D          ;PUT SP ON STACK
1089 C703 3A1BD8        LDA      ALDC
1090 C706 67            MOV      H,A
1091 C707 3A20D8        LDA      FLDC
1092 C70A 6F            MOV      L,A
1093 C70B E5            PUSH     H          ;PUT A + FLAGS ON STACK
1094 C70C 3A21D8        LDA      ILDC
1095 C70F 67            MOV      H,A
1096 C710 E5            PUSH     H          ;PUT INT. MASK ON STACK
1097 C711 211CD8        LXI      H,BLDC
1098 C714 56            MOV      D,M
1099 C715 23            INX      H
1100 C716 5E            MOV      E,M
1101 C717 23            INX      H
1102 C718 D5            PUSH     D          ;PUT B-C ON STACK
1103 C719 56            MOV      D,M
1104 C71A 23            INX      H
1105 C71B 5E            MOV      E,M
1106 C71C D5            PUSH     D          ;PUT D-E ON STACK
1107 C71D 23            INX      H
1108 C71E 23            INX      H

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1109 C71F 23          INX      H
1110 C720 56          MOV      D,M
1111 C721 23          INX      H
1112 C722 5E          MOV      E,M
1113 C723             ;
1114 C723             ;RESTORE CPU REGISTERS WITH USER DATA.
1115 C723             ;
1116 C723 EB          XCHG          ;H-L SET
1117 C724 D1          POP      D      ;D-E SET
1118 C725 C1          POP      B      ;B-C SET
1119 C726 F1          POP      PSW
1120 C727             +      SIM
      C727 30         +      DB      30H
      C728
1121 C728 F1          POP      PSW      ;A + FLAGS SET
1122 C729 2232D8      SHLD     TL0C+10 ;STORE H-L TEMP
1123 C72C E3          XTHL          ;SP IN H
1124 C72D F9          SPHL          ;SP LOADED
1125 C72E 2A34D8      LHLD     EXIT    ;GET JUMP ADDR.
1126 C731 E5          PUSH     H      ;PUT ON NEW STACK
1127 C732 2A32D8      LHLD     TL0C+10 ;SET H-L
1128 C735 FB          EI
1129 C736 C9          RET              ;EXECUTE USER CODE.
1130 C737             ;
1131 C737             ;G02 SET BREAKPOINT
1132 C737             ;
1133 C737 DAFC06      G02:      JC      G01      ;NO BREAKPOINT TO SET.
1134 C73A 0E2D        MVI      C,'-'      ;SENT PROMPT
1135 C73C CDB1C2      CALL     OUTC      ;FOR BREAKPOINT
1136 C73F CD33C4      CALL     GET16     ;GET BREAKPOINT ADDR.
1137 C742 2239D8      SHLD     TL0C      ;STORE FIRST BREAKPOINT ADDR.
1138 C745 7E          MOV      A,M      ;GET CODE FROM PROG.
1139 C746 322AD8      STA      TL0C+2    ;SAVE
1140 C749 36C7        MVI      M,0C7H    ;SET BREAKPOINT #1
1141 C74B DA64C7      JC      G03
1142 C74E 0E2D        MVI      C,'-'
1143 C750 CDB1C2      CALL     OUTC
1144 C753 CD33C4      CALL     GET16     ;GET ANOTHER ADDR
1145 C756 2239D8      SHLD     TL0C+3
1146 C759 7E          MOV      A,M      ;GET CODE FROM PROG
1147 C75A 322DD8      STA      TL0C+5    ;SAVE
1148 C75D 36C7        MVI      M,0C7H    ;SET BREAKPOINT #2
1149 C75F 3EFF        MVI      A,-1
1150 C761 3231D8      STA      TL0C+9
1151 C764 210000      G03:      LXI      H,0
1152 C767 112ED8      LXI      D,TL0C+6
1153 C76A 7E          MOV      A,M
1154 C76B 12          STAX     D
1155 C76C 13          INX      D
1156 C76D 23          INX      H
1157 C76E 7E          MOV      A,M
1158 C76F 12          STAX     D
1159 C770 23          INX      H
1160 C771 13          INX      D
1161 C772 7E          MOV      A,M

```


SPACE BYTE 3035 FROM MONITOR 11-14-77

```

1162 C773 12          STAX    D
1163 C774 210000      LXI     H,0
1164 C777 36C3       MVI     M,003H
1165 C779 2132C7      LXI     H,BREAK
1166 C77C 220100      SHLD    1
1167 C77F C3F0C6      JMP     GO1      ;RESTORE REGS, AND JUMP.
1168 C782            ;
1169 C782            ;BREAK RETURN WHEN A BREAKPOINT IS HIT
1170 C782            ;
1171 C782 F3          BREAK: DI          ;DISABLE THE INTERRUPT
1172 C783 E3          XTHL
1173 C784 2B          DEX     H
1174 C785 2234D8      SHLD    EXIT      ;SAVE THE NEXT USER ADDR
1175 C788 23          INX     H
1176 C789 E3          XTHL
1177 C78A E5          PUSH    H
1178 C78B D5          PUSH    D
1179 C78C C5          PUSH    B
1180 C78D F5          PUSH    PSW
1181 C78E            +      RIM
1182 C78E 20          +      DB      20H
1183 C78F            ;
1183 C78F            ;STORE REGISTERS IN SAVE AREA IN SCRATCHRAM
1184 C78F            ;
1185 C78F 2121D8      LXI     H,ILDC
1186 C792 77          MOV     M,A
1187 C793 211BD8      LXI     H,ALDC
1188 C796 C1          POP     B
1189 C797 70          MOV     M,B
1190 C798 2120D8      LXI     H,FLDC
1191 C79B 71          MOV     M,C
1192 C79C 2110D8      LXI     H,BLDC
1193 C79F C1          POP     B
1194 C7A0 70          MOV     M,B
1195 C7A1 23          INX     H
1196 C7A2 71          MOV     M,C
1197 C7A3 23          INX     H
1198 C7A4 C1          POP     B
1199 C7A5 70          MOV     M,B
1200 C7A6 23          INX     H
1201 C7A7 71          MOV     M,C
1202 C7A8 C1          POP     B
1203 C7A9 2123D8      LXI     H,LLDC
1204 C7AC 71          MOV     M,C
1205 C7AD 2B          DEX     H
1206 C7AE 70          MOV     M,B
1207 C7AF C1          POP     B
1208 C7B0 2124D8      LXI     H,PLDC
1209 C7B3 70          MOV     M,B
1210 C7B4 23          INX     H
1211 C7B5 71          MOV     M,C
1212 C7B6 23          INX     H
1213 C7B7 EB          XCHG
1214 C7B9 210000      LXI     H,0

```

SPACE BYTE 3085 FROM MONITOR 11-14-77

```

1215 07BB 39          DAD      SP
1216 07BC EB          XCHG
1217 07BD 72          MOV      M,D
1218 07BE 23          INX      H
1219 07BF 73          MOV      M,E
1220 07C0 31FFD8      LXI      SP,STACK
1221 07C3              ;
1222 07C3              ;EVERYTHING NOW STORED RESTORE
1223 07C3              ;BREAKPOINT DATA
1224 07C3              ;
1225 07C3 2A28D8      LALD      TLDC
1226 07C6 3A2AD8      LDA      TLDC+2
1227 07C9 77          MOV      M,A
1228 07CA 3A31D8      LDA      TLDC+9
1229 07CD A7          ANA      A
1230 07CE CAD8C7      JZ       B1
1231 07D1 2A2BD8      LALD      TLDC+3
1232 07D4 3A2DD8      LDA      TLDC+5
1233 07D7 77          MOV      M,A
1234 07D8 97          SUB      A
1235 07D9 3231D8      STA      TLDC+9
1236 07DC 112ED8      LXI      D,TLDC+6
1237 07DF 210000      LXI      H,0
1238 07E2 1A          LDAX     D
1239 07E3 77          MOV      M,A
1240 07E4 13          INX      D
1241 07E5 23          INX      H
1242 07E6 1A          LDAX     D
1243 07E7 77          MOV      M,A
1244 07E8 13          INX      D
1245 07E9 23          INX      H
1246 07EA 1A          LDAX     D
1247 07EB 77          MOV      M,A
1248 07EC              ;
1249 07EC              ;EXIT THRU R TO MAIN.
1250 07EC              ;
1251 07EC 21B0C6      LXI      H,RTBL
1252 07EF C38EC6      JMP      R5
1253 07F2              ;
1254 07F2              ;POP PSW AND ENTER HERE
1255 07F2              ;
1256 07F2 F5          VIDEO:  PUSH   PSW
1257 07F3 C5          PUSH   B
1258 07F4 D5          PUSH   D
1259 07F5 E5          PUSH   H
1260 07F6 47          MOV      B,A
1261 07F7 3A38D8      VID1:  LDA      IFL
1262 07FA FE49          CPI      'I'      ;FLAG INITED?
1263 07FC CA11C8      JZ       CONT      ;YES, DISPLAY CHAR
1264 07FF AF          INI:    XRA      A      ;CLEAR A
1265 0800 D3F8          OUT      0F8H      ;ZERO SCROLL
1266 0802 2138D8      LXI      H,IFL
1267 0805 3649          MVI      M,'I'      ;SET IFL
1268 0807 2C          INR      L
1269 0808 3630          MVI      M,30H

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

1270	C80A	2C		INR	L	
1271	C80B	2C		INR	L	
1272	C80C	3600		MVI	M,0	
1273	C80E	C8B0C3		JMP	CLEAR	
1274	C811					
1275	C811	2A36D8	CONT:	LHLD	CURS	
1276	C814	3A3AD8		LDA	UND	
1277	C817	77		MOV	M,A	
1278	C818	78		MOV	A,B	
1279	C819	B7		ORA	A	
1280	C81A	CA8FC3		JZ	DELAY	
1281	C81D	FE13		CPI	13H	;CNTRL S 'SPEED'
1282	C81F	CAE0C3		JZ	SDL	
1283	C822	FE15		CPI	21	;CNTRL U 'CURSUR UP'
1284	C824	CA96C3		JZ	UP	
1285	C827	FE12		CPI	18	;CNTRL R 'CURSUR FORWARD'
1286	C829	CA5DC3		JZ	FOR	
1287	C82C	FE0C		CPI	12	;CNTRL L 'BACKSPACE'
1288	C82E	CADCC3		JZ	BACK	
1289	C831	FE0E		CPI	14	;CNTRL N 'GRAPHICS ON'
1290	C833	CA83C3		JZ	GRON	
1291	C836	FE0F		CPI	15	;CNTRL O 'GRAPHICS OFF'
1292	C838	CA83C3		JZ	GROFF	
1293	C83B	FE04		CPI	4	;CNTRL D 'CLEAR SCREEN'
1294	C83D	CAFFC7		JZ	INI	
1295	C840	FE0A		CPI	10	;CNTRL J 'LINE FEED'
1296	C842	CAD3C3		JZ	LF	
1297	C845	FE0D		CPI	13	;CNTRL M 'RETURN'
1298	C847	CAC2C3		JZ	CR	
1299	C84A	FE08		CPI	8	;CNTRL H 'HOME CURSUR'
1300	C84C	CABCC3		JZ	HOME	
1301	C84F	FE5F		CPI	95	;SHIFT Q 'BACKSPACE'
1302	C851	CADAC3		JZ	BS	
1303	C854	E67F		ANI	7FH	
1304	C856	47		MOV	B,A	
1305	C857	3A39D8		LDA	VFL	
1306	C85A	B0		ORA	B	
1307	C85B	47		MOV	B,A	
1308	C85C	70		MOV	M,B	
1309	C85D	23	FOR:	INX	H	
1310	C85E	7C	DONE:	MOV	A,H	
1311	C85F	FEFC		CPI	PSTAD+4	
1312	C861	C830C3		JNZ	FINI	
1313	C864	2100F8		LXI	H,PAGE	
1314	C867	1140F8		LXI	D,PAGE+64	
1315	C86A	1A	SCROL:	LDAX	D	
1316	C86B	77		MOV	M,A	
1317	C86C	13		INX	D	
1318	C86D	23		INX	H	
1319	C86E	7A		MOV	A,D	
1320	C86F	FEFC		CPI	PSTAD+4	
1321	C871	C86AC3		JNZ	SCROL	
1322	C874	36A0	LDL:	MVI	M,0A0H	
1323	C876	23		INX	H	
1324	C877	7C		MOV	A,H	

SPACE BYTE 8085 PROM MONITOR 11-14-77

1325	C878	FEFC		CPI	PSTAD+4
1326	C87A	C274C8		JNZ	LDL
1327	C87D	21C0FB		LXI	H,PAGE+960
1328	C880	7E	FINI:	MOV	A,M
1329	C881	323AD8		STA	UND
1330	C884	36FF		MVI	M,-1
1331	C886	00		NOP	
1332	C887	2236D8		SHLD	CURS
1333	C88A	E1		POP	H
1334	C88B	D1		POP	D
1335	C88C	C1		POP	B
1336	C88D	F1		POP	PSW
1337	C88E	C9		RET	
1338	C88F	3C	DELAY:	INR	A
1339	C890	C29FC8		JNZ	DELAY
1340	C893	C380C8		JMP	FINI
1341	C896	11C0FF	UP:	LXI	D,-64
1342	C899	19		DAD	D
1343	C89A	7C	FIX:	MOV	A,H
1344	C89B	E603		ANI	3
1345	C89D	F6FB		ORI	PSTAD
1346	C89F	67		MOV	H,A
1347	C8A0	C380C8		JMP	FINI
1348	C8A3	3E80	GROFF:	MVI	A,80H
1349	C8A5	C3AAC8		JMP	VSET
1350	C8A8	3E00	GRON:	MVI	A,0
1351	C8AA	3239D8	VSET:	STA	VFL
1352	C8AD	C380C8		JMP	FINI
1353	C8B0	2100FB	CLEAR:	LXI	H,PAGE
1354	C8B3	36A0	CLEAR1:	MVI	M,0A0H
1355	C8B5	23		INX	H
1356	C8B6	7C		MOV	A,H
1357	C8B7	FEFC		CPI	PSTAD+4
1358	C8B9	C2B3C8		JNZ	CLEAR1
1359	C8BC	2100FB	HOME:	LXI	H,PAGE
1360	C8BF	C380C8		JMP	FINI
1361	C8C2	7D	CR:	MOV	A,L
1362	C8C3	E6C0		ANI	0C0H
1363	C8C5	6F		MOV	L,A
1364	C8C6	3A3BD8		LDA	SPD
1365	C8C9	57		MOV	D,A
1366	C8CA	7A	DLY:	MOV	A,D
1367	C8CB	A7		ANA	A
1368	C8CC	1B		DCX	D
1369	C8CD	C2CAC8		JNZ	DLY
1370	C8D0	C380C8		JMP	FINI
1371	C8D3	114000	LF:	LXI	D,64
1372	C8D6	19		DAD	D
1373	C8D7	C35EC8		JMP	DONE
1374	C8DA	36A0	BS:	MVI	M,0A0H
1375	C8DC	2B	BACK:	DCX	H
1376	C8DD	C39AC8		JMP	FIX
1377	C8E0	3A3BD8	SDL:	LDA	SPD
1378	C8E3	C614		ADI	20
1379	C8E5	323BD8		STA	SPD

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1380 C8E8 C380C3          JMP      FINI
1381 C8EB                ;
1382 C8EB                ;DISK BOOTSTRAP SECTION
1383 C8EB                ;
1384 C8EB CDB0C1      BBOOT:  CALL    ECHO
1385 C8EE FE0D                CPI     13
1386 C8F0 C222C4                JNZ    ERROR
1387 C8F3 CD4AC5                CALL   CRLF
1388 C8F6                ;
1389 C8F6 DBCC          BOOT:   IN     PORTD
1390 C8F9 A7              ANA     A
1391 C8F9 C204C9                JNZ    BOOTT
1392 C8FC C217C2                LXI    H,MS014
1393 C8FF 1E07              MVI    E,SZ014
1394 C901 C327C4                JMP    CHK1
1395 C904                ;
1396 C904 31FFD8      BOOTT:  LXI    SP,STACK
1397 C907 CD17C9                CALL   BOOTT1
1398 C90A C34000                JMP    40H
1399 C90D                ;
1400 C90D 3E81          RESET:  MVI    A,81H
1401 C90F CDE1CA                CALL   LOOP
1402 C912 3E0D              MVI    A,13
1403 C914 C3E1CA                JMP    LOOP
1404 C917                ;
1405 C917 CD0DC9      BOOTT1:  CALL   RESET
1406 C91A 210000                LXI    H,0      ;SET OFFSET = 0
1407 C91D 65              PUSH    H
1408 C91E 21F401                LXI    H,500    ;SIZE OF DISK EXECUTIVE
1409 C921 220CD8                SHLD   ISIZE    ;STORE IN INPUT FILE SIZE REG.
1410 C924 210ED8                LXI    H,ITRK   ;POINT TO INPUT FILE TRACK REG.
1411 C927 3601              MVI    M,1      ;STORE TRACK 1
1412 C929 23              INX     H
1413 C92A 3600              MVI    M,0      ;STORE SECTOR 0
1414 C92C 23              INX     H
1415 C92D 3600              MVI    M,0      ;STORE INPUT COUNT = 0
1416 C92F CD4CC9                CALL   LOADER   ;LOAD THE DISK EXECUTIVE
1417 C932 C9              RET
1418 C933                ;
1419 C933 2A17D8      RESTR:  LHLD    TISZ    ;GET ORIGINAL SIZE OF FILE
1420 C936 220CD8                SHLD   ISIZE    ;COPY TO WORKING SIZE REGISTER
1421 C939 3A16D8                LDA     TITRK   ;GET ORIGINAL TRACK NUMBER
1422 C93C 320ED8                STA     ITRK    ;COPY TO WORKING TRACK REGISTER
1423 C93F 3A0BD8                LDA     IUNIT
1424 C942 0F              RRC
1425 C943 0F              RRC
1426 C944 320FD8                STA     ISCTR
1427 C947 97              SUB     A
1428 C948 3210D8                STA     IONTR
1429 C94B C9              RET
1430 C94C                ;
1431 C94C                ;LOAD ROUTINE TO READ A HEX OBJECT FILE FROM DISK
1432 C94C                ;AND STORE IN MAIN RAM MEMORY.
1433 C94C                ;
1434 C94C E1          LOADER:  POP     H      ;GET OFFSET AND SWAP

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1435 C94D E3          XTHL          ;WITH RETURN
1436 C94E E5          PUSH          H
1437 C94F E1          LOAD: POP      H
1438 C950 E5          PUSH          H
1439 C951 CDC1C9      CALL          RIX
1440 C954 063A        MVI           B, 'A'
1441 C956 90          SUB           B          ;LOOK FOR DATA LINE.
1442 C957 C24FC9      JNZ          LOAD
1443 C95A 57          MOV           D, A          ;STORE IN D
1444 C95B CD90C9      CALL          BYTE
1445 C95E CA81C9      JZ          LOAD2
1446 C961 5F          MOV           E, A
1447 C962 CD90C9      CALL          BYTE
1448 C965 F5          PUSH          PSW
1449 C966 CD90C9      CALL          BYTE
1450 C969 C1          POP           B
1451 C96A 4F          MOV           C, A
1452 C96B 09          DAD           B
1453 C96C CD90C9      CALL          BYTE
1454 C96F CD90C9      LOAD1: CALL     BYTE
1455 C972 77          MOV           M, A
1456 C973 23          INX           H
1457 C974 1D          DCR           E
1458 C975 C26FC9      JNZ          LOAD1
1459 C978 CD90C9      CALL          BYTE
1460 C97B C2A7C9      JNZ          LODER
1461 C97E C34FC9      JMP          LOAD
1462 C981 CD90C9      LOAD2: CALL     BYTE
1463 C984 67          MOV           H, A
1464 C985 CD90C9      CALL          BYTE
1465 C988 6F          MOV           L, A
1466 C989 B4          ORA           H
1467 C98A CA8EC9      JZ          LOAD3
1468 C98D E9          PCHL
1469 C98E E1          LOAD3: POP      H
1470 C98F C9          RET
1471 C990          ;
1472 C990 CDC1C9      BYTE: CALL     RIX
1473 C993 CDAFC9      CALL          NBL
1474 C996 07          RLC
1475 C997 07          RLC
1476 C998 07          RLC
1477 C999 07          RLC
1478 C99A 4F          MOV           C, A
1479 C99B CDC1C9      CALL          RIX
1480 C99E CDAFC9      CALL          NBL
1481 C9A1 B1          ORA           C
1482 C9A2 4F          MOV           C, A
1483 C9A3 82          ADD           D
1484 C9A4 57          MOV           D, A
1485 C9A5 79          MOV           A, C
1486 C9A6 C9          RET
1487 C9A7          ;
1488 C9A7          ;LODER IF LOAD ERROR OCCURS
1489 C9A7          ;

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1490 C9A7 21F5C1    LODER: LXI      H,MS002 ;ERROR MESSAGE
1491 C9AA 1E0E      MVI      E,SZ002 ;SIZE TO E
1492 C9AC C327C4    JMP      CHK1
1493 C9AF          ;
1494 C9AF          ;SUBR TO CONVERT AN ASCII CHARACTER
1495 C9AF          ;TO A 4-BIT BINARY VALUE
1496 C9AF          ;
1497 C9AF D630      NBL:      SUI      '0'
1498 C9B1 D8        RC
1499 C9B2 C6E9      ADI      0E9H
1500 C9B4 D8        RC
1501 C9B5 C606      ADI      6
1502 C9B7 F2BDC9    JP      N10
1503 C9BA C607      ADI      7
1504 C9BC D8        RC
1505 C9BD C60A      N10:      ADI      10
1506 C9BF B7        ORA      A
1507 C9C0 C9        RET
1508 C9C1          ;
1509 C9C1          ;SUBROUTINE TO READ A ASCII CHAR FROM DISK
1510 C9C1          ;CHARACTER RETURNED IN A.
1511 C9C1          ;
1512 C9C1 CDD4C9    RIX:      CALL     RI
1513 C9C4 DA2EC1      JC      MAIN
1514 C9C7 E67F      ANI      127
1515 C9C9 C9        RET
1516 C9CA          ;
1517 C9CA          ;CASIN READ A BYTE FROM MAG TAPE
1518 C9CA          ;
1519 C9CA DB6E      CASIN:    IN      CASC ;GET MAG TAPE STATUS
1520 C9CC E610      ANI      16 ;CHECK B4
1521 C9CE C2CAC9    JNZ      CASIN ;WAIT TIL LD
1522 C9D1 DB6F      IN      CASC ;GET BYTE
1523 C9D3 C9        RET      ;RETURN TO CALLER
1524 C9D4          ;
1525 C9D4          ;RI READ AN 8 BIT BYTE FROM DISKETTE
1526 C9D4          ;
1527 C9D4 C5      RI:      PUSH     B
1528 C9D5 E5      PUSH     H
1529 C9D6 2110D8    LXI      H,ICNTR ;GET INPUT FILE COUNT
1530 C9D9 7E      MOV      A,M
1531 C9DA A7      ANA      A ;SET FLAGS
1532 C9DB C23ECA    JNZ      RI10 ;IF NOT 0 THEN BRANCH
1533 C9DE 210FD8    RI5:      LXI      H,ISCTR ;COUNT IS 0, INCREMENT SECTOR
1534 C9E1 CD53CA    CALL     INCD8
1535 C9E4 2A0CD8    LALD     ISIZE ;GET INPUT SIZE
1536 C9E7 2B      DCR      H ;DECREMENT BY ONE
1537 C9E9 220CD8    SHLD     ISIZE ;SECTOR
1538 C9EB 7D      MOV      A,L ;CHECK IF OVER MAX
1539 C9EC A7      ANA      A
1540 C9ED C202CA    JNZ      RI3 ;NOT TO MAX YET!
1541 C9F0 7C      MOV      A,H
1542 C9F1 A7      ANA      A ;TEST HI BYTE
1543 C9F2 C202CA    JNZ      RI3
1544 C9F5          ;

```

SPACE BYTE 9085 FROM MONITOR 11-14-77

```

1545 C9F5      ;WE RAN OVER INDICATED SIZE WITHOUT
1546 C9F5      ;FINDING END OF FILE CHARACTER.  EXIT
1547 C9F5      ;WITH CARRY SET INDICATING ERROR.
1548 C9F5      ;
1549 C9F5 23      INX      H
1550 C9F6 220CD8   SHLD     ISIZE
1551 C9F9 2110D8   LXI      H,ICNTR
1552 C9FC 3600     MVI      M,0
1553 C9FE 37      STC
1554 C9FF E1      R12:    POP      H
1555 CA00 C1      POP      B
1556 CA01 C9      RET
1557 CA02 210FD8   R13:    LXI      H,ISCTR
1558 CA05 CDBCCA   CALL     XUS
1559 CA08 CD08CB   CALL     CHK
1560 CA0B 23      INX      H
1561 CA0C 3680     MVI      M,128
1562 CA0E 0E05     MVI      C,5
1563 CA10 210ED8   LXI      H,ITRK
1564 CA13 CDD2CA   CALL     SEEK
1565 CA16 3E03     R16:    MVI      A,3
1566 CA18 CDE1CA   CALL     LOOP
1567 CA1B DBCC     IN       PORTD
1568 CA1D E680     ANI      80H
1569 CA1F CA28CA   JZ       R14
1570 CA22 CDDFCA   CALL     RFLAG
1571 CA25 C3DEC9   JMP      R15
1572 CA28 DBCC     R14:    IN       PORTD
1573 CA2A E608     ANI      8
1574 CA2C CA3ECA   JZ       R110
1575 CA2F CDDFCA   CALL     RFLAG
1576 CA32 0D      DCR      C
1577 CA33 C218CA   JNZ      R16
1578 CA36 2103C2   LXI      H,MS003
1579 CA39 1E0F     MVI      E,SZ003
1580 CA3B C327C4   JMP      CHK1
1581 CA3E      ;
1582 CA3E 3E40     R110:   MVI      A,40H
1583 CA40 D3D9     OUT      PORTA
1584 CA42 DBCC     IN       PORTD
1585 CA44 4F      MOV      C,A
1586 CA45 3E41     MVI      A,41H
1587 CA47 CDE1CA   CALL     LOOP
1588 CA4A 2110D8   LXI      H,ICNTR
1589 CA4D 35      DCR      M
1590 CA4E 79      MOV      A,C
1591 CA4F B7      ORA      A
1592 CA50 C3FFC9   JMP      R12
1593 CA53      ;
1594 CA53      ;ROUTINE TO INCREMENT DISK ADDRESS
1595 CA53      ;
1596 CA53 34      INCDR:   INR      M
1597 CA54 7E      MOV      A,M
1598 CA55 E61F     ANI      1FH
1599 CA57 FE1B     CPI      27

```


SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1600 CA59 C45E0A      JZ      INCDB
1601 CA5C 2D          DCR      L
1602 CA5D C9          RET
1603 CA5E 7E          INCDB:  MOV    A,M
1604 CA5F E6C1        ANI      0C1H
1605 CA61 77          MOV      M,A
1606 CA62 2D          DCR      L
1607 CA63 34          INR      M
1608 CA64 C9          RET
1609 CA65            ;
1610 CA65            ;SUBROUTINE TO WRITE A BYTE TO CASSETTE TAPE
1611 CA65            ;
1612 CA65 D36E        CASQT:  IN      CASC      ;GET CASSETTE STATUS
1613 CA67 E620        ANI      20H      ;LOOK AT B6
1614 CA69 C2650A      JNZ      CASQT      ;WAIT TIL LD
1615 CA6C 79          MOV      A,C      ;GET CHAR
1616 CA6D D36F        OUT      CASC      ;SEND TO CASSETTE
1617 CA6F C9          RET
1618 CA70            ;
1619 CA70            ;
1620 CA70            ;SUBROUTINE TO WRITE A BYTE TO DISK
1621 CA70            ;
1622 CA70 79          WRT:    MOV      A,C
1623 CA71 E5          PUSH     H
1624 CA72 D3DA        OUT      PORTB
1625 CA74 3E31        MVI      A,31H
1626 CA76 CDE10A      CALL     LOOP
1627 CA79 2115D3      LXI      H,DCNTR
1628 CA7C 34          INR      M
1629 CA7D 7E          MOV      A,M
1630 CA7E FE90        CPI      128
1631 CA80 C2BACA      JNZ      WRT4
1632 CA83 3600        MVI      M,0
1633 CA85 2114D3      WRT1:    LXI      H,DSCTR
1634 CA88 CD2CCA      CALL     KUS
1635 CA8B CD08CB      CALL     CHK
1636 CA8E 0E05        MVI      C,5
1637 CA90 2B          DCR      H
1638 CA91 CDD2CA      CALL     SEEK
1639 CA94 3E05        WRT2:    MVI      A,5
1640 CA96 CDE10A      CALL     LOOP
1641 CA99 3E07        MVI      A,7
1642 CA9B CDE10A      CALL     LOOP
1643 CA9E DBCC        IN      PORTD
1644 CAA0 E608        ANI      8
1645 CAA2 CAB7CA      JZ      WRT3
1646 CAA5 CDDFCA      CALL     RFLAG
1647 CAA8 0D          DCR      C
1648 CAA9 C294CA      JNZ      WRT2
1649 CAAC 3E0F        MVI      A,15
1650 CAAE CDE10A      CALL     LOOP
1651 CAB1 CDEDCA      CALL     WRTN
1652 CAB4 C385CA      JMP      WRT1
1653 CAB7 CDEDCA      WRT3:    CALL     WRTN
1654 CABA E1          WRT4:    POP      H

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1655 CAB8 09          RET
1656 CAB0             ;
1657 CAB0             ;SUBROUTINE TO TRANSMIT UNIT SECTOR BYTE
1658 CAB0             ;
1659 CAB0 7E          XUS:  MOV      A,M
1660 CABD E61F         ANI      1FH
1661 CABF E5          PUSH     H
1662 CAC0 213BCB       LXI      H,TBL-1
1663 CAC3 85          ADD      L
1664 CAC4 6F          MOV      L,A
1665 CAC5 4E          MOV      C,M
1666 CAC6 E1          POP      H
1667 CAC7 7E          MOV      A,M
1668 CAC8 E6C0         ANI      0C0H
1669 CACA B1          ORA      C
1670 CACB D3DA       XXUS:  OUT      PORTB
1671 CACD 3E21        MVI      A,21H
1672 CACF C3E1CA       JMP      LOOP
1673 CAD2             ;
1674 CAD2             ;SUBROUTINE TO SEEK TRACK
1675 CAD2             ;TRACK NUMBER IS IN A
1676 CAD2             ;
1677 CAD2 7E          SEEK:  MOV      A,M
1678 CAD3 D3DA       OUT      PORTB
1679 CAD5 3E11        MVI      A,11H
1680 CAD7 CDE1CA       CALL     LOOP
1681 CADA 3E09        MVI      A,9
1682 CADC C3E1CA       JMP      LOOP
1683 CADF             ;
1684 CADF             ;SUBROUTINE TO RESET DISK FLAGS
1685 CADF             ;
1686 CADF 3E0B        RFLAG:  MVI      A,11
1687 CAE1             ;
1688 CAE1             ;SUBROUTINE TO ISSUE COMMAND..
1689 CAE1             ;THEN LOOP ON BUSY UNTIL DONE.
1690 CAE1             ;
1691 CAE1 D3D9        LOOP:   OUT      PORTA
1692 CAE3 97          SUB      A
1693 CAE4 D3D9        OUT      PORTA
1694 CAE6 DBCC        LOOP1:  IN       PORTD
1695 CAE8 1F          RAR
1696 CAE9 DAE6CA       JC       LOOP1
1697 CAEC 09          RET
1698 CAED             ;
1699 CAED             ;SUBR TO INC DISK ADDR AND
1700 CAED             ;CHECK OUTPUT FILE SIZE.
1701 CAED             ;
1702 CAED 2114D8       WRTN:   LXI      H,DSCTR
1703 CAF0 CD53CA       CALL     INCDA
1704 CAF3 2A11D8       WRTN2:  LHLD     DSIZE
1705 CAF6 2B          DCR      H
1706 CAF7 2211D8       SHLD     DSIZE
1707 CAF9 7D          MOV      A,L
1708 CAFB A7          ANA      A
1709 CAFC C0          RNZ

```


SPACE BYTE 3035 FROM MONITOR 11-14-77

```

1710 CAFD 7C          MOV      A,H
1711 CAFE A7          ANA      A
1712 CAFF C0          RNZ
1713 CB00 2112C2       LXI      H,MS004
1714 CB03 1E09        MVI      E,SZ004
1715 CB05 C327C4       JMP      CHK1
1716 CB08             ;
1717 CB08             ;SUBROUTINE TO CHECK IF DISKETTE IS PRESENT
1718 CB08             ;
1719 CB08 DBCC        CHK:    IN      PORTD
1720 CB0A E620         ANI      20H
1721 CB0C C9          RZ
1722 CB0D 211BC2       LXI      H,MS005
1723 CB10 1E0E        MVI      E,SZ005
1724 CB12 C327C4       JMP      CHK1
1725 CB15             ;
1726 CB15             ;INTERPASS FUNCTIONS
1727 CB15             ;
1728 CB15 3A08D8       IPASS:  LDA      PASS
1729 CB18 1F          RAR
1730 CB19 D228CB       JNC      PASS2
1731 CB1C 3A08D8       LDA      PASS
1732 CB1F 3D          DCR      A
1733 CB20 3208D8       STA      PASS
1734 CB23 3E01        MVI      A,1
1735 CB25 C339C9       JMP      PASS3
1736 CB28 A7          PASS2:  ANA      A
1737 CB29 CA48C0       JZ       UPDAT
1738 CB2C CD33C9       CALL    RESTR
1739 CB2F 3A08D8       LDA      PASS
1740 CB32 1F          RAR
1741 CB33 F5          PUSH     PSW
1742 CB34 97          SUB      A
1743 CB35 3208D8       STA      PASS
1744 CB38 F1          POP      PSW
1745 CB39 C630        PASS3:  ADI      30H
1746 CB3B C9          RET
1747 CB3C             ;
1748 CB3C             ;LOGICAL/PHYSICAL SECTOR TABLE
1749 CB3C             ;
1750 CB3C 010A1302     TBL:    DB      1,10,19,2
1751 CB40 0B14030C     DB      11,20,3,12
1752 CB44 15040D16     DB      21,4,13,22
1753 CB48 050E1706     DB      5,14,23,6
1754 CB4C 0F180710     DB      15,24,7,16
1755 CB50 1908111A     DB      25,3,17,26
1756 CB54 0912         DB      9,13
1757 CB56             ;
1758 CB56             ;TAPEL ROUTINE TO READ TARBELL TAPE
1759 CB56             ;RECORD INTO RAM, OR COMPARE TAPE TO RAM.
1760 CB56             ;
1761 CB56 F5          TAPEL:  PUSH     PSW      ;SAVE THE CONTROL CHAR
1762 CB57 DB6E         IN      CASC      ;SEE IF CASSETTE
1763 CB59 FEFF         CPI      -1       ;IS INSTALLED.
1764 CB5B CAF3C2       JZ       WT2      ;IT IS NOT!

```

SPACE BYTE 8085 FROM MONITOR 11-14-77

1765 CB5E 3E10	MVI	A,10H	;RESET
1766 CB60 D36E	OUT	CASC	;CASSETTE INTERFACE
1767 CB62 97	SUB	A	;CLEAR
1768 CB63 3206D3	STA	ERBYT	;WRITE ERROR REG.
1769 CB66 CD12C4	CALL	STRING	;GET SA AND EA
1770 CB69 D1	POP	D	;ENDING ADDRESS
1771 CB6A E1	POP	H	;STARTING ADDRESS
1772 CB6B F1	POP	PSW	;SWAP STARTING CHAR
1773 CB6C E5	PUSH	H	;WITH STARTING ADDR.
1774 CB6D F5	PUSH	PSW	;ON STACK.
1775 CB6E 0600	MVI	B,0	;SET CHECKSUM = 0
1776 CB70 CDCAC9	TAP1: CALL	CASIN	;GET A CHAR FROM TAPE
1777 CB73 4F	MOV	C,A	;SAVE IN REG. C
1778 CB74 F1	POP	PSW	;LOOK AT CONTROL CHAR.
1779 CB75 F5	PUSH	PSW	
1780 CB76 FE56	CPI	'V'	;VERIFY ONLY?
1781 CB78 CA7CCB	JZ	TAP2	;YES!
1782 CB7B 71	MOV	M,C	
1783 CB7C 79	TAP2: MOV	A,C	
1784 CB7D 80	ADD	B	;UPDATE
1785 CB7E 47	MOV	B,A	;CHECKSUM
1786 CB7F 79	MOV	A,C	;CHECK IF BYTE
1787 CB80 BE	CMP	M	;WRITTEN PROPERLY
1788 CB81 C4C3CB	CNZ	TAP4	;KEEP TRACK OF ERRORS
1789 CB84 CD08C4	CALL	LIMIT	;SEE IF WERE DONE.
1790 CB87 D270CB	JNC	TAP1	;GET MORE INPUT
1791 CB8A CDCAC9	CALL	CASIN	;GET CHECKSUM
1792 CB8D B8	CMP	B	;COMPARE TO CALCULATED VALUE
1793 CB8E CA99CB	JZ	TAP3	;GOOD LOAD
1794 CB91 21F5C1	LXI	H,MS002	
1795 CB94 1E0E	MVI	E,SZ002	
1796 CB96 C327C4	JMP	CHK1	
1797 CB99 3A06D8	TAP3: LDA	ERBYT	
1798 CB9C A7	ANA	A	
1799 CB9D CAB1CB	JZ	TAP6	
1800 CBA0 CD4AC5	CALL	CRLF	
1801 CBA3 3A06D8	LDA	ERBYT	
1802 CBA6 CD77C4	CALL	DIS8	
1803 CBA9 2152C2	LXI	H,MS010	
1804 CBAC 1E14	MVI	E,SZ010	
1805 CBAE C32AC4	JMP	CHK1+3	
1806 CBB1 F1	TAP5: POP	PSW	;GET CONTROL CHAR
1807 CBB2 FE56	CPI	'V'	
1808 CBB4 CAC3CB	JZ	TAP5	
1809 CBB7 FE4C	CPI	'L'	
1810 CBB9 CA2EC1	JZ	MAIN	
1811 CBBC FE45	CPI	'E'	
1812 CBBE C2C3CB	JNZ	TAP4	
1813 CBC1 E1	POP	H	
1814 CBC2 E9	POHL		
1815 CBC3 2166C2	TAP5: LXI	H,MS011	
1816 CBC6 1E0D	MVI	E,SZ011	
1817 CBC8 C327C4	JMP	CHK1	
1818 CBCB E5	TAP4: PUSH	H	
1819 CBCC 2106D8	LXI	H,ERBYT	

SPACE BYTE 8085 FROM MONITOR 11-14-77

```

1320 CBCF 34          INR      M
1321 CBD0 E1          POP      H
1322 CBD1 C9          RET
1323 CBD2              ;
1324 CBD2              ;WTAPE WRITE A MAG TAPE BINARY FILE
1325 CBD2              ;
1326 CBD2 DB6E        WTAPE: IN      CASC      ;CHECK TO SEE IF
1327 CBD4 FEFF        CPI      -1          ;MAG TAPE IS INSTALLED
1328 CBD6 CAF9CB      JZ       WT2         ;ITS NOT!
1329 CBD9 CD13C4      CALL     STRING      ;GET START AND END ADDR.
1330 CBDC D1          POP      D          ;HI ADDR
1331 CBDD E1          POP      H          ;START ADDR.
1332 CBDE 0600        MVI      B,0         ;ZERO CHECKSUM
1333 CBE0 CD65CA      CALL     CASQT       ;SCRATCH CHAR
1334 CBE3 0EE6        MVI      C,0EE6H    ;SEND
1335 CBE5 CD65CA      CALL     CASQT       ;SYNC CHAR
1336 CBE8 4E          WT1:  MOV      C,M     ;GET BYTE FROM MEMORY
1337 CBE9 CD65CA      CALL     CASQT       ;WRITE TO TAPE
1338 CBEC 80          ADD      B          ;UPDATE
1339 CBED 47          MOV      B,A        ;CHECKSUM
1340 CBEE CD08C4      CALL     LIMIT      ;CHECK IF DONE
1341 CBF1 D2E9CB      JNC      WT1        ;LOOP IF NOT
1342 CBF4 48          MOV      C,B        ;SEND
1343 CBF5 C365CA      JMP      CASQT      ;CHECKSUM TO TAPE
1344 CBF8              ;
1345 CBF8 217AC2      WT2:  LXI      H,MS015
1346 CBF8 1E0B        MVI      E,SZ015
1347 CBFD C327C4      JMP      CHK1
1348 CC00              ;
1349 CC00 0000        END

```

TOTAL ERRORS=00

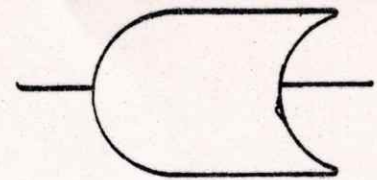
SPACE BYTE 3035 FROM MONITOR 11-14-77

ADIS1	L	C638	ADISP	L	C629	ALOC	L	D81B	AS1	L	C625
ASSEM	L	C036	ASSIG	L	C603	B1	L	C7D8	BACK	L	C8DC
BAUDC	L	D819	BAUDP	L	D804	BBOOT	L	C8EB	BLOC	L	D81C
BOUT	L	C8F6	BOUT1	L	C917	BOUTT	L	C904	BREAK	L	C782
BS	L	C8DA	BYTE	L	C990	BYTTP	L	C581	CASC	E	006E
CASD	E	006F	CASIN	L	C9CA	CASOT	L	CA65	CHK	L	CB08
CHK1	L	C427	CLEAR	L	C8B3	CLEAR	L	C8B0	CLDC	L	D81D
CONT	L	C911	CR	L	C8C2	CRL1	L	C2A8	CRLF	L	C54A
CRLFD	L	C1EA	CRLFP	L	C299	CRLFV	L	C033	CSTAT	L	D807
CURS	L	D836	DEL1	L	C463	DELAY	L	C88F	DELIM	L	C460
DISB	L	C477	DISAD	L	C46F	DLAY	L	C376	DLDC	L	D81E
DLOOP	L	C37C	DLY	L	C8CA	DONE	L	C85E	DREG	L	C6AA
DUMP	L	C55D	DUMP1	L	C562	DUMP2	L	C568	ECHO	L	C1B0
EDITO	L	C03F	ELDC	L	D81F	ERBYT	L	D806	ERROR	L	C422
EXIT	L	D834	EXIT1	L	C3BD	EXIT2	L	C3C3	FILL	L	C528
FILL1	L	C532	FINI	L	C880	FIX	L	C89A	FLAGV	L	C027
FLDC	L	D820	FOW	L	C85D	GET0	L	C439	GET1	L	C43C
GET16	L	C433	GET2	L	C44F	GO	L	C6ED	GO1	L	C6FC
GO2	L	C737	GO3	L	C764	GROFF	L	C8A3	GRON	L	C8A8
HEXAS	L	C3E9	HIBYT	L	D801	HILDC	L	C3B0	HLOC	L	D822
HOME	L	C8BC	HTDA	L	C499	ICNTR	L	D810	IFL	L	D838
ILDC	L	D821	INC	L	C387	INC1	L	C39A	INC2	L	C38E
INCDA	L	CA53	INCDB	L	CA5E	INCV	L	C003	INI	L	C7FF
INIT	L	C057	INIT0	L	C119	INIT1	L	C079	INIT2	L	C095
INIT3	L	C0BD	INIT4	L	C0CA	INIT5	L	C0DA	INIT6	L	C0E5
INIT7	L	C0F0	INIT8	L	C0FB	INIT9	L	C0B2	INITA	L	C106
INITB	L	C08C	INITC	L	C11C	INITD	L	C111	INSP1	L	C5E4
INSP2	L	C5FF	INSPE	L	C5DE	IPASS	L	CB15	ISCTR	L	D80F
ISIZE	L	D80C	ITRK	L	D80E	IUNIT	L	D80B	LOCCN	L	C457
LDL	L	C874	LF	L	C8D3	LIMIT	L	C408	LIST	L	C489
LLDC	L	D823	LOAD	L	C94F	LOAD1	L	C96F	LOAD2	L	C981
LOAD3	L	C98E	LOADE	L	C94C	LODER	L	C9A7	LOOP	L	CAE1
LOOP1	L	CAE6	LOOP2	L	C122	LOOPV	L	C02A	MAIN	L	C12E
MEMSZ	L	C018	MEMTP	L	C3E1	MLDC	E	D822	MODE	E	4000
MOVE1	L	C541	MOVEB	L	C53A	MS001	L	C1BD	MS002	L	C1F5
MS003	L	C203	MS004	L	C212	MS005	L	C21B	MS006	L	C229
MS007	L	C239	MS008	L	C23C	MS009	L	C243	MS010	L	C252
MS011	L	C266	MS013	L	C276	MS014	L	C273	MS015	L	C27A
MS016	L	C235	MS017	L	C295	MSOUT	L	C3A6	N10	L	C9BD
NBL	L	C9AF	NULL	L	C1A2	NULDC	L	D800	QCNT	L	D815
QFILE	L	D809	QSCTR	L	D814	QSIZE	L	D811	QTRK	L	D813
QUNIT	L	D80A	QUTC	L	C2B1	QUTC1	L	C2BC	QUTCV	L	C009
QUTP	L	C593	QUTP1	L	C59E	QUTP2	L	C59C	QUTP3	L	C5BD
QUTP4	L	C5BF	QUTPV	L	C015	PAGE	E	F800	PASS	L	D808
PASS2	L	CB28	PASS3	L	CB39	PASSV	L	C030	PLDC	L	D824
PORTA	E	00D9	PORTB	E	00DA	PORTC	E	00DB	PORTD	E	00DC
PORTI	E	00D8	PPROG	L	C31B	PPULS	E	0010	PRNAD	L	C520
PROG	L	C051	PROG1	L	C32C	PROG2	L	C332	PROGA	L	C352
PROM	L	C2C7	PROM1	L	C2D2	PROM2	L	C30D	PSTAD	E	00F8
PSTAT	L	D803	R	L	C640	R0	L	C64B	R1	L	C65A
R2	L	C65F	R3	L	C67F	R4	L	C680	R5	L	C68E
R6	L	C691	READP	E	0024	READV	L	C006	RESET	L	C90D
RESTR	L	C933	RESTV	L	C01B	RFLAG	L	CADF	RI	L	C9D4
RI10	L	CA3E	RI2	L	C9FF	RI3	L	CA02	RI4	L	CA28
RI5	L	C9DE	RI6	L	CA16	RIM	M	7C40	RIV	L	C00F

SPACE BYTE 8085 FROM MONITOR 11-14-77

RIX	L	C9C1	RPRM	L	C2E3	RSTRV	L	C02D	RTBL	L	C6BC
RKRDY	E	0002	S0110	E	46D1	S0150	E	4500	S0300	E	4290
S1200	E	40A0	S1300	E	406B	S2400	E	4050	S4300	E	4028
S9600	E	4014	SCRDL	L	C36A	SDATA	E	00C0	SDL	L	C3E0
SEK	L	CAD2	SEKVV	L	C024	SIM	M	7C31	SLDC	L	D826
SP10	L	C514	SP11	L	C4EC	SP2	L	C4F7	SP3	L	C4C3
SP4	L	C51C	SP5	L	C4CE	SP6	L	C4D9	SP7	L	C503
SP9	L	C503	SPD	L	D83B	SPEED	L	C4A8	SSTAT	E	00C1
STACK	E	D8FF	START	L	C000	STB	E	0001	STRIN	L	C412
STRN0	L	C41F	SZ001	E	002D	SZ002	E	000E	SZ003	E	000F
SZ004	E	0009	SZ005	E	000E	SZ006	E	0010	SZ007	E	0003
SZ008	E	0007	SZ009	E	000F	SZ010	E	0014	SZ011	E	000D
SZ013	E	0004	SZ014	E	0007	SZ015	E	000B	SZ016	E	0010
SZ017	E	0004	SZR	E	000C	TAP1	L	C370	TAP2	L	C37C
TAP3	L	C399	TAP4	L	C3C3	TAP5	L	C3C3	TAP6	L	C3B1
TAPEL	L	C356	TBL	L	C33C	TEMP	L	D83C	TIMHI	E	00DD
TIMLO	E	00DC	TISZE	L	D817	TITRK	L	D816	TLDC	L	D828
TWRTV	L	C00C	TXRDY	E	0001	UND	L	D83A	UP	L	C396
UPDAT	L	C043	VFL	L	D839	VID1	L	C7F7	VIDEO	L	C7F2
VSET	L	C8AA	WRT	L	CA70	WRT1	L	CA85	WRT2	L	CA94
WRT3	L	CAB7	WRT4	L	CABA	WRTN	L	CAED	WRTN2	L	CAF3
WRTV	E	000E	WRTV	L	C012	WT1	L	CBE3	WT2	L	CBF8
WTAPE	L	CBD2	XUS	L	CABC	XUSV	L	C01E	XXUS	L	CACB
XXUSV	L	C021									

WARRANTY REGISTRATION



MUST BE MAILED WITHIN 10 DAYS OF PURCHASE

When this SPACE BYTE-8085 CPU was released, it was believed to be free of any design and/or production flaws. However, it is possible that through our continuing aggressive engineering and reevaluation, combined with user feedback, modifications might become advisable.

So that we may keep you advised in this event, please fill out and return this registration form. Please verify the SPACE BYTE serial number of your board at the top of this form.

NAME: _____

ADDRESS: _____

CITY & STATE: _____ ZIP _____

PHONE: _____

PURCHASED FROM: _____

DISK TYPE: _____

* * *

SPACE BYTE
WARRANTY DISCLOSURE

The specific warranty period for the SPACE BYTE 8085 CPU is 90 days from the date of purchase verified by receipt of buyers registration form at SPACE BYTE, Los Angeles, California. During that period the device will be repaired free of charge if material, component or manufacturing defects become apparent. However your dealer should be consulted prior to contacting SPACE BYTE. Most actual device replacements can be handled quickly by the dealer from who you purchased your unit.

This warranty is void if the SPACE BYTE has been subjected to catastrophic electrical or mechanical abuse. In this case the unit will be repaired for a nominal fee to cover actual device replacement and a labor charge. This charge will be authorized in writing prior to commencement of repairs.

NOTICE: The Space Byte Corporation hereby assumes no liability for loss of data, system damage or personal injury resulting from the use of said product.

The SPACE BYTE Corporation
1720 Pontius ave. Suite 201
Los Angeles, Calif. 90025
(213) 468-8080

